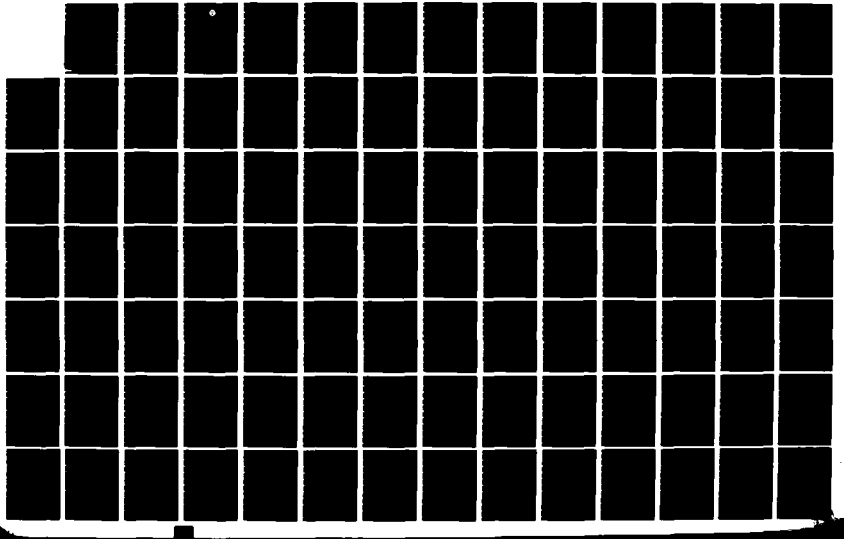
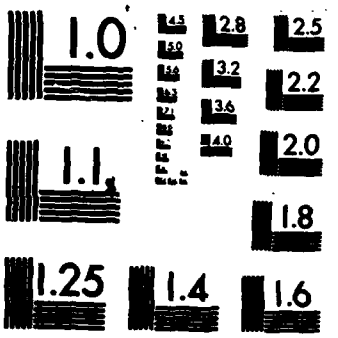
AD-A137 541 ACTIVE RELIABILITY ENGINEERING - TECHNICAL CONCEPT AND PROGRAM PLAN A SOL..(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA D D HALL 05 OCT 83 NOSC/TD-654 1/2

UNCLASSIFIED F/G 9/2 NL





NOSC TD 654

NOSC TD 654

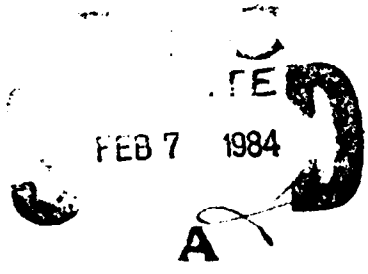
AD A137541

Technical Document 654

ACTIVE RELIABILITY ENGINEERING — TECHNICAL CONCEPT AND PROGRAM PLAN

A solid-state systems approach to increased reliability and availability in military systems

D. D. Hall



5 October 1983
Period of Work:
May — October 1982

Approved for public release; distribution unlimited.

DTIC FILE COPY

NOSC

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

NAVAL OCEAN SYSTEMS CENTER SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

J.M. PATTON, CAPT, USN
Commander

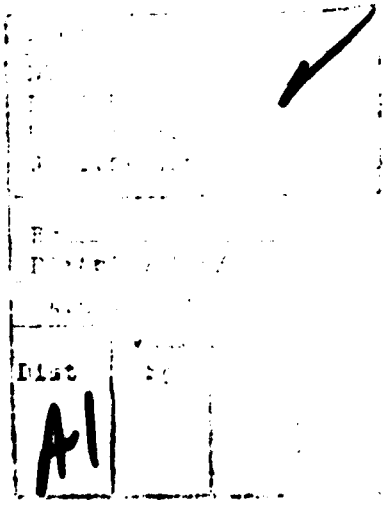
R.M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

This work, supported by proposal funds from the Engineering and Computer Sciences Directorate, Code 09, was performed by a member of the Testing Technology Office (Code 9304) from May to October 1982. This document was approved for publication 5 October 1983.

Released by
M.E. Nunn, Head
Testing Technology Office

Under authority of
C.L. Ward, Head
Product Engineering Department



WR

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Document 654 (TD 654)	2. GOVT ACCESSION NO. AD-A13754/	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ACTIVE RELIABILITY ENGINEERING –TECHNICAL CONCEPT AND PROGRAM PLAN A solid-state systems approach to increased reliability and availability in military systems		5. TYPE OF REPORT & PERIOD COVERED Final May–October 1982 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) D.D. Hall		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Proposal funds
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 5 October 1983 13. NUMBER OF PAGES 102
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fault tolerance
Reliability engineering
System architecture

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A technical program plan for a technology base development of a self-repair concept for complex digital systems is presented in this document. The underlying technical concept provides for bridging the gap between traditional fault tolerance and standard passive reliability engineering techniques by the coupling of statistical techniques to the application of redundancy and use of intelligent control circuitry for switching between redundant units. This technique can provide significant increases in the statistical availability of Navy systems at reasonable increase in cost, even though it may not prove absolute tolerance of one or

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Continued

more faults. A program plan is presented for the development of this technology base as well as application to specific areas of interest, including multicomputer networks and VLSI/VHSIC integrated circuits.

EXECUTIVE SUMMARY

This document represents the culmination of an effort to define a technology base and an applications-oriented program in the general subject area of fault tolerance. This effort, supported by proposal funds from the Engineering and Computer Sciences Directorate at NOSC, Code 09, was 4 months in duration. Current technical literature was studied. Ongoing government-funded R&D programs, industry-funded development activities, and industry independent research and development (IRAD) projects were reviewed. Existing and developmental systems in the Fleet were examined to uncover maintenance and reliability issues that could be addressed by this subject. From these studies, an assessment of the state of the art of fault tolerance was established and played against an analysis of Fleet requirements. By examining the resulting requirements matrix, technological deficiencies were uncovered. A solution to these issues was conceptualized and formulated into a technology base related to fault tolerance but distinct in several important points. Applications for this novel technology were sought and described. Finally a detailed program to develop this technology was defined and a resulting program plan was laid out.

From this analysis, it was determined that

- Most research activity in fault tolerance has taken place outside of DOD in response to different requirements.
- Many DOD fault tolerant technology programs "tag along" to work being done by programs outside DOD. Also, the work funded by DOD is fragmented and does not address uniform goals.
- The objective of those programs is to effect an extremely high probability of failure-free mission of given duration, to the severe discount of other specification parameters such as acquisition cost and life-cycle maintenance.
- With one or two isolated exceptions, the technical concept to achieve fault tolerance in industry usually is based on massive added redundancy achieved at considerable added acquisition costs.
- Manufacturing technology has progressed much more rapidly than fault tolerance technology. Thus, in many cases, a reliability goal is met by improved manufacturing techniques sooner than by a fault tolerance program.
- A significant gap in realized reliability exists between the point where conventional reliability techniques leave off and fault tolerance takes over. A technology to fill this gap is badly needed.
- This gap can be filled by a new approach consisting of selective use of redundancy based on statistical modeling mediated by intelligent error detecting and control circuitry.

- The redundancy inherent in the "network" approach to design in the development of many new pieces of equipment and systems makes it a natural for adaptation to this new approach at reasonable cost.

- An aggressive directed research and development program is needed to bring this new tech base to realization on a generally applicable basis.

Traditional fault tolerance is based on equal application of redundancy to all system components. However, its effectiveness is limited to the extent it enhances the reliability of the least reliable component. If the differential between least reliable and most reliable components is large, then systematic redundancy is largely wasted. This is the source of disinterest evidenced by DOD program managers in the use of fault tolerance as an effective means of extending system reliability. The replacement of blanket large-scale redundancy by selective application of redundant components on a statistically derived basis, coupled with the replacement of simple voter-based decision logic by intelligent redundancy control circuitry, results in a new approach to reliability enhancement. This approach can rival or surpass the enhancement obtainable by traditional fault tolerance when considered on a statistical basis (overall availability), and at much reduced cost. It can "guarantee" operation only for failure modes most likely to occur; it cannot guarantee operation in spite of all possible failure modes. This concept is the subject of this paper.

The need for a reliability extending technology as a tool for the system designer is apparent. This tool can be used most effectively to reduce life-cycle cost by reducing unscheduled maintenance. It also can be used to increase system availability or unsupported mission duration. These advantages are being recognized by various isolated programs usually under the label fault tolerance for lack of another name. A notable example is the AN/UYK-43 computer development program. The name proposed herein for this novel technology is "active reliability engineering." These isolated efforts lack having a sufficiently developed technology base to draw upon. This proposed program is defined for the purpose of filling this deficiency.

Active reliability is applicable to all levels of system design, from major multinode command control systems through shipboard combat systems down to individual VHSIC integrated circuit chips. It is especially attractive to systems based on a network design concept. The network approach consists of integrating a number of identical or similar components around a data transfer mechanism. This concept is also being applied at all levels of system design. The network approach with its inherent redundancy is ideally suited to active reliability approaches. All that is required to make the network actively reliable is the addition of intelligent control circuitry.

This document describes the active reliability concept as an alternative to traditional fault tolerance. It also describes several types of applications. One key application is the network of embedded computers. Networks of this type are being proposed as a solution to a variety of system design problems. VHSIC technology is another key application. In both cases, the complexity of the resulting system makes reliability enhancement an attractive

added feature. A comprehensive program plan covering several funding categories is provided herein. It is recommended that this program be initiated and vigorously prosecuted to make this technology base available to system development managers.

CONTENTS

ILLUSTRATIONS

1 INTRODUCTION

No responsible engineer designs electronic circuits to be prone to failure. No responsible system designer deliberately structures a system so that component failures have a high probability of propagating their effects throughout the greater system, thereby causing catastrophic system failure. Nevertheless, components do fail, on occasion, and these failures do cause circuit malfunctions, on occasion. In addition, circuit malfunctions do drive the overall system to an unavailable (failed) state, on occasion. It is an engineering project management fact of life that the number of occasions of individual component failure throughout the life of a piece of equipment is an inverse function of the amount of engineering effort (and funding) devoted to failure reduction.

Traditional reliability-centered approaches to this problem focus on design and production techniques devised to prevent failures of components. The practice of conservative design—design for wide performance and support (e.g., cooling) margins, simplicity of design, component selection, over-specification of components, screening and burn-in of incoming and outgoing products—is a well-established and accepted process for reducing the number of component failures. These techniques also cost time, money, and effort in the design and production phases of the equipment acquisition process. Generally, the budget and time period for these activities are limited, thus component failures are allowed. In many cases, they occur uncomfortably often, especially after the equipment has been placed into service. The use of traditional reliability techniques to avoid failures in equipment can be described as passive reliability.

The passive approach to reliability—traditional reliability engineering—is a mature and stable engineering discipline. Its techniques and concepts are well documented in military standards and handbooks. However, in some cases, system requirements for reliability have exceeded the capability of traditional techniques. The passive approach will fall short in some cases for the following reasons: (1) The application of reliability engineering becomes increasingly costly as the desired failure rate (subject to the law of diminishing returns) is reduced. (2) There are practical theoretical limits to the degree to which component failure rates can be reduced. (3) Equipment using state-of-the-art components suffer from the lack of experimentally defined reliability characteristics. (4) As a consequence of its complexity modern integrated circuitry is expensive to screen and test relative to its cost. (5) There is no way to perform reliability prediction and detailed reliability testing on integrated circuitry.

As a result of these factors, system designers, in some cases, have chosen to supplement the passive approach to reliability with untraditional techniques. These approaches are characterized as active, as opposed to passive. They consist of including as part of system design the ability to detect component failures (or their resulting effects on circuits) and to reconfigure faulty systems so that they continue to operate correctly in spite of one or more component failures. Although many names have been advanced to describe this approach, the most common and best accepted is fault tolerance. The goal of fault tolerance as a design discipline is to greatly increase the mission mean-time-between-failure (MTBF) statistic, the most important reliability parameter, by designing circuits and systems so that a component failure

(i.e., fault) does not render the overall circuit or system inoperative. Note that fault tolerant design is predicated on the premise that a fault will inevitably happen.

Fault tolerance as a design discipline consists of preventing the propagation of a fault or its effects and substituting correctly operating (or operable) circuitry for the circuit containing the faulty component(s). This process requires some enhancements over the basic (non-fault-tolerant) equipment design. Additional capability must be added to detect faults. In addition, spare (redundant) circuitry must be included. Finally, very reliable decision, control, and switching circuitry must be included. This all adds to the final weight, space, and cost of the equipment.

Historically, fault tolerance has been achieved by brute force redundant techniques. In early applications to analog or mechanical systems, the entire systems were triplicated or quadruplicated from end to end (control to actuator) to achieve the required degree of failure protection. An example is control of the flight control surfaces in commercial aircraft. The same approach was applied to computer systems when, initially, reliability needed improvement. Computers were triplicated, and all three identical computer units were run synchronously in parallel. A "voting circuit" would decide which was the failed computer by accepting the result of the two units that agreed in their output. Note that the brute force techniques provide protection against all faults except those within the voter, but at a high cost—over three times that of the original computer. In addition, although the mission reliability (the length of time from start-up with all three computers running fault-free to system failure, with two out of three failed) is greatly increased, the long-term reliability (the inverse of the number of failures over a period of time) is decreased. There is three times as much equipment to fail.

Some later applications of fault tolerance used newer techniques that emphasized total engineering of the system and its components to achieve an overall active reliability goal. Reliance on the simple brute force approach to fault tolerance was abandoned and replaced by an approach that included engineering each of the subunits of the system to contribute to an active reliability concept. Thus, components were modified rather than simply replicated "as is" (without reengineering). Considerable redundancy is still required, but less than before. In addition, the use of error detecting and correcting codes as an active reliability tool was introduced. A good example is the telephone company's electronic switching system (ESS).

Strange as it may seem, still newer techniques are returning to the emphasis of large-scale redundancy, primarily as a means of achieving even greater reliability. Triplication with voting is the basis for two proposed solutions to the problem of ultrareliable digital flight control systems for commercial aircraft. In addition, these more recent techniques provide for identical spare units (four and above the basic three) to allow functional replacement of one of the three and thus to further increase mission reliability. Again, a significant cost increase is encountered to achieve these levels of reliability. As the technology of fault tolerance is pushed to applications requiring extreme reliability, the gap widens between the point at which passive reliability leaves off and fault tolerance takes over.

The brute force approaches used for the application of fault tolerance and the resulting costs involved have tended to restrict and limit its applicability. Fault tolerance is commonly viewed as an all-or-nothing proposition to be used where extreme mission reliability is mandatory and its cost therefore justified. Otherwise, the standard passive techniques are necessarily considered adequate in the absence of cost-effective alternatives. Thus applications of fault tolerance in the past have been limited in number. This technology has not been brought to the full maturity exemplified by passive reliability because interest has been somewhat limited. There has been only a limited amount of work done to advance the general state of the art of fault tolerance. Because of these facts, applications have been approached on a case-by-case basis, each application utilizing a unique technique. As such, with a few exceptions, a designers bag of standard practice "tricks" has not evolved. There are no standard procedures for applying fault tolerance in general to equipment and systems. There are neither standard techniques nor an accepted technical language for trading off fault tolerance with passive reliability to achieve by their combination a given overall reliability figure.

The problem of language is significant; it results from the fact that passive reliability and fault tolerance have pursued seemingly diverse objectives. Passive reliability seeks to avoid failures, and its success is measured in terms of the statistical time that a failure is avoided (MTBF). Fault tolerance concedes the possibility of a failure and measures its success in terms of the ability to avoid the effects of failures (i.e., it looks at the types and number of failures that can be tolerated before loss of a system occurs). These are different terms, depicting different concepts. One is viewed narrowly as taking off where the other ends. Thus, the ability to trade the two off against each other to achieve a combined goal is lacking. Uniform language for achieving a specification for total (active and passive) reliability does not exist.

In addition, the existing reliability models, having been derived for a strictly passive approach, are not adequate for a combined approach. Without good models, there can be no systematic basis for cost and benefit trade-offs between the active and passive techniques. The ability to tolerate failures during a mission is only a secondary parameter of interest. What the system specifier is really interested in is the mission reliability figure, which is expressed in terms of either the statistical measure of the time into a mission before encountering a system-killing failure or, alternatively, the probability of encountering a system-killing failure during a mission of given duration. Thus the ability to tolerate failures comes into play only where it is impossible or too costly to avoid the failures. A uniform cost trade-off model would provide for cost and benefit measures over a continuum of active and passive alternatives.

Fault tolerance has tended to be an extreme deterministic concept involving absolute tolerance of one or more faults. Passive reliability has tended to be a statistical quantity involving the probability of success or the statistical time to failure. What is needed is a concept and technology that fills the middle ground between the two. This concept would be by definition a statistically based active reliability technology. It would also fill in a gap in application. This concept would fill the middle ground between the extreme reliability that can be provided by fault tolerance and

the limitations imposed by passive reliability. It would extend reliability to meet requirements beyond the capability of passive reliability on a compatible statistical basis, but would demonstrate much more cost effectiveness than strict fault tolerance. The development of this technology is the goal of this program, and its title, "Active Reliability Engineering," reflects its affinity with passive practice.

Fault tolerance has tended to be an extreme measure (all or nothing) also because system designers were not able to get into the design of individual subelements of the system. For example, the triple redundancy approach modifies the system at the system level (by adding a voter box) but does not impact the design of each computer. Even in cases in which the system designer was able to get into the individual subelement designs, this process was avoided for the sake of simpiicity of design thought. There are exceptions (e.g., ESS), but brute force thinking still is a major factor driving fault tolerance design experts.

A related problem is that of multitiered requirements. Requirements for reliability can change over the duration of a mission. For example, shipboard weapon and electronic systems can be expected to be involved in missions lasting 60 days or more. However, extreme fault tolerance may be required only for some critical functions, and then for a much abbreviated period (say 24 hours) of high battle readiness. The ability to tier the implementation of active reliability from the extreme requirement down to more relaxed requirements over a long mission will achieve a much more cost-effective solution for these requirements. This is especially true if the human-in-the-loop is taken into account. In fact, extreme fault tolerance for the totality of shipboard systems will be cost prohibitive in the near future. Some brute force approaches render future maintenance and repair even more difficult by masking system faults from the maintenance staff.

The goal of this proposed program is to establish the technical discipline of active reliability engineering. Although the definition of active reliability includes fault tolerance, this program will pursue active reliability engineering as contrasted to brute force fault tolerance. The technology of brute force approaches is being investigated elsewhere. Active reliability engineering is defined as the engineering of the total system to include active measures to support a system reliability goal, including the judicious use of redundancy on a statistical basis. There are two parts to this definition, both of which emphasize the contrast with fault tolerance. The first part is total engineering of the system, especially the engineering of system subelements, to support active reliability. The engineering of system subelements down to the level of the integrated circuit component would indeed help to solve many of the problems now associated with fault tolerance. This concept would allow capture of the advantages of integrated circuitry by impacting the designs of individual integrated circuits. There are other advantages as well. The main one is that a relatively modest support function for active reliability can be designed into the subelement, to be built up with other capabilities in other subelements to achieve considerable fault tolerance at the system level. Considerable fault tolerance is achieved when subelement functions are combined with system-level functions. Low-end active reliability is achieved mostly by the subelement functions themselves. Thus the solution can be tailored to match the requirement.

This approach (active reliability engineering) is not the same as engineering each subelement to be fault tolerant by itself, then building a fault tolerant system out of fault tolerant subelements. The theory behind the concept proposed herein is that a basic active reliability capability designed into the subelement will be less complex and costly than the capability required to make each subelement fault tolerant. While the determination of the exact nature of this basic capability is the goal, this program will consist in general of three functions: (1) Aid in detecting and isolating fault conditions. (2) Control of the subelement (switched in or out; turned on or off) by the system. (3) Inclusion of limited redundancy within the subelement. Within the scope of this concept, a requirement may still exist for some limited application of fault tolerance to the subelement to support a multitiered system reliability. However, the concept does not simply remove brute force fault tolerance from the system and place it in the subelement.

The second part of this definition for active reliability engineering is the use of selective redundancy as statistically required, as opposed to blanket triplication. Without bringing the statistical basis of passive reliability into the active reliability engineering process, this process can not be cost effective. Brute force fault tolerance triplicates every circuit component. It would be applied equally to a highly reliable integrated circuit, such as a microprocessor that would not be likely to fail over the expected life of the system, as well as to a power amplifier that may fail every 100 hours. Why triplicate the first example with identical circuitry likely never to be called on? Why on the other hand only triplicate a subelement that will fail repeatedly over a typical mission? Blind fault tolerance does both. Unfortunately, active reliability engineering must overcome absolute fault tolerance concepts to become accepted because it does allow some faults to disrupt the system. The theory is that it allows only very rare ones to do so. This is not absolute fault tolerance but can be a clear extension over present passive reliability. Active reliability models are required that support this statistical design process.

The time for this program is most opportune. In fact, it is critical from the standpoint of the present state of technology. One reason is the transition from LSI to VLSI and VHSIC integrated circuit technology. This technology, in pressing the physical limits associated with the physics of semiconductor material, sets limits on passive reliability. In addition, this new technology will see more use made of "custom" integrated circuit devices (designed for a particular application or class of applications) as opposed to "off the shelf" devices (general purpose). It is easier to design these features into a custom device than into a general purpose one.

The fact that more custom devices will be used will necessitate a greater dependence on computer-aided design (CAD) automated design aids. The existence and widespread use of these tools provide an ideal facility for institutionalizing standard active reliability design concepts into subelement design processes without encountering the lengthy learning curve normally associated with such advances. Finally, there is considerable industry interest, as evidenced by commercial products in development and by industry independent research and development (IRAD). This work needs to be captured

for military applications by evaluating industry outputs and promulgating direction concerning Navy requirements for ongoing industrial work.

This program is concerned with studying active reliability more than simply for its own sake. Specific impacts are anticipated provided associated programs are funded. These outputs also serve to characterize the progress of the program as it pursues this technology to its full conclusion of seeing active reliability design factors fully embedded in all phases of equipment acquisition. The payoff of the program does not lie solely with its full conclusion. The interim impacts are key, both to system designers that can make use of them and as milestones to the long-term goal. These milestones are described later in this document.

A program in active reliability engineering could also be referred to as active maintainability engineering. The ability of a system itself to maintain its operation automatically has to be included when the role of maintenance personnel is evaluated. Additionally, the problems of repair of a system while it operates (to be called on-line maintenance) and the added maintenance burden created by an active concept must be addressed. The system maintenance philosophy and level-of-repair analysis must become integral elements in the active reliability engineering process. Active reliability potentially impacts virtually all phases of maintenance in both a positive and negative way. It will impact the ability to test and check out equipment, since active reliability requires fault detection and tends to mask faults. If a system must meet a requirement for very high availability, the most effective method of providing it might be to combine active reliability with repair in place (on-line maintenance)—i.e., repair while the system continues doing its job. This philosophy must be incorporated into the maintenance approach for the system. Alternatively, a maintenance requirement for a maintenance-free mission of several months will place constraints on the active reliability requirements and approach.

This program is not about fault tolerance per se, as fault tolerance is commonly understood. Fault tolerance is somewhat esoteric to the system designer and in theory tends to address the upper end of active reliability (total coverage, zero latency). Thus, its application has tended to be quite limited. The proposal herein is intended to be more far-reaching both in its cost-benefit practicality and in application of standardized processes and techniques via universal CAD tools. It also approaches the problem from a probabilistic point of view compatible with passive reliability rather than strict absolute tolerance of faults. In light of these comments, the title chosen for this program reflects some divergence from traditional fault tolerance and from association with passive reliability.

2 PROGRAM OVERVIEW

DEFINITION OF ACTIVE RELIABILITY ENGINEERING

In today's technical terminology, reliability engineering refers to what has been called passive reliability. This term has been defined as the set of design and production techniques used to render a system resistant to faults by making its components less prone to faults. The goal of this engineering discipline is to avoid the statistical occurrence of a fault in the system over a specified interval of time. It consists of two fully developed and stable subdisciplines: managerial and technical. The technical discipline consists of the design and production engineering procedures. It is somewhat of a "bottom up" technical philosophy. Systems are rendered reliable by engineering their subelements for reliability, subelements are made reliable by means of their immediate components, and so on down to the individual component level. The management discipline consists of those processes that establish a reliability program, monitor the analysis and design processes during the program execution, and provide for formal verification tests. Management discipline is institutionalized in various military standards and handbooks and can be invoked for any system acquisition effort. Passive reliability has universal applicability to all types and sizes of equipment and to all levels of system design, because of its bottom-up approach.

Although the use of active techniques would be a natural extension of the use of passive reliability techniques, a great dichotomy exists between the relative states of practice of the two. Active reliability has been defined as the use of some activity on the part of the system when in service to maintain its correct operation, even if it encounters random failures in its components. It is important to note the differences today between passive and active (mostly as evidenced by fault tolerance) reliability. As mentioned, passive reliability is a bottom-up design philosophy, whereas fault tolerance is top-down. Fault tolerance is applied to a system as a whole by adjusting the design of the system with minor alteration to its subelements. In its most brute force case (triplication) the subelements are unchanged from their design for the basic (non-fault-tolerant) system. In more sophisticated cases, their design is altered slightly. Active reliability is inherently a system or top-down philosophy because it must span a certain breadth of components to be effective. Active reliability can become more bottom-up oriented if general functions that the components must provide to the system as a whole are identified. This top-down nature prevents fault tolerance from having the universal applicability of passive reliability. In fact, it is quite specialized to the certain applications in which it has been required. The only way to avoid this problem is for one program to look at the widespread general application of active reliability rather than at a solution to a particular problem.

The directed goal of this program is to advance the discipline of active reliability engineering and elevate it to the point at which it approximates the general applicability enjoyed by passive reliability. The ideal goal in the long term would be to have a single universal reliability engineering discipline that combines both passive and active procedures. This long-term goal can only be approximated in the near term. Passive reliability is a universal discipline applicable to all phases of equipment design and should continue to be mandated for all system developments. Significant improvements

in user-experienced reliability can be achieved if active reliability measures are employed at a level of universality approaching that of passive reliability, but they should be employed only as the standard practice of active reliability engineering is made available. The title of this program is indicative of this goal. The key to attainment of this objective is the establishment of procedures, functional definitions, and design techniques whereby the subelements of a system (at all levels of system definition) support active reliability for the system as a whole; i.e., active reliability engineering will be more "bottom up" than will straight fault tolerance.

The active reliability engineering process also consists of a management discipline and a technical discipline. The technical discipline consists of (1) A methodical process for deriving meaningful active reliability requirements from a single reliability specification. (2) An orderly process for developing reliability design budgets and reliability values for subelement specifications. (3) The ability to apply standard design practices for active reliability to the system as a whole and functional decomposition procedures to apply those procedures to the subelement. (4) The ability to design to these functional requirements. (5) The ability to conduct uniform meaningful validation and verification procedures. This process will become applicable to systems of all sizes.

The management discipline consists of (1) Definition of uniform procedures for specifying combined reliability levels. (2) Definition of the procedures for prosecuting a management program for active reliability, including formal design reviews to track the analysis and design process. (3) Institutionalization of standard design practice rules and functions via handbooks and design guides. (4) Functional specification of universal active reliability functions. The management discipline is as important as the technical discipline to the widespread application of active reliability engineering. Widespread use of the management discipline is critical to the success of active reliability. The approach underlying this program addresses both aspects.

ACTIVE RELIABILITY ENGINEERING IN THE VLSI ENVIRONMENT

Active reliability is both desirable and essential as a standard design discipline to be incorporated in all future system designs. Although not every system will have the same degree of active reliability, every system designer should have the procedure for specifying and implementing a variety of mature (he does not have to develop them) active reliability techniques to be used in conjunction with standard passive practice to achieve an overall multitiered reliability goal. To the question, "Why should we develop active reliability as a standard design discipline, as opposed to simply making further advances in the area of passive reliability?" the answer is simply that (1) passive reliability has gone about as far as it can go as a standard practice, and (2) active techniques are becoming increasingly cost competitive. Both parts to this answer center around the existence of highly integrated microminiature electronic circuitry.

The effect of increasing degrees of integration is to allow the placement of increasingly higher degrees of circuit complexity in a single package (higher functional density). This process of design both aids and limits the effectiveness of passive reliability to meet a given goal. It is an aid

because a higher degree of circuit integration reduces the component count and number of interconnections for a given level of functional complexity, thus enhancing reliability. A large scale integration (LSI) implementation of a computer will be more reliable than an MSI implementation of the exact same (functional) computer. That's good for passive reliability. On the other hand larger scales of integration make possible the inclusion of more complex functions, negating much of the advantage just mentioned. Of more importance, highly integrated circuits reach upper limits on the level of reliability to which they can be passively engineered. This is because each higher level of integration more fully utilizes the material of a small block of semiconductor material and thus is more prone to defects and flaws that can appear in the material due to a variety of physical causes (e.g., radiation). Furthermore it is difficult to "derate" an integrated circuit, as one would derate discrete components, while retaining the same level of integration (i.e., maintaining the same functional complexity). This problem becomes critical as physical limits to the level of miniaturization are approached. It is especially critical today as the transition from large scale integration (LSI) to very large scale integration (VLSI) and very high speed integrated circuits (VHSIC) takes place.

However, an increasing degree of circuit integration also allows the incorporation of additional capabilities (such as those required to support active reliability) at relatively low cost, provided that these capabilities are designed into the integrated circuit device itself. This statement implies that the subunits of the overall system and the individual microcircuits themselves must be engineered for active reliability. Thus the increase in functional complexity required to implement an active technique can be achieved at incremental cost, whereas large relative cost increases will be encountered in achieving incremental gains in passive reliability.

ACTIVE RELIABILITY—AN EXAMPLE

A good existing example of the active reliability engineering process in action is the use of memory error detecting and correcting (EDAC) codes. Such codes exist in variety and are used to correct single or double errors in a memory array by the storage of an encoded version of each memory space. These codes are implemented with a relatively small increase in circuitry (i.e., 30% as opposed to 300%) to achieve the same overall goal of single or double fault tolerance. However, memory EDAC is one of very few examples of active reliability that is in standard use today. Codes of this type for random logic are being investigated by university research but are not yet in common use. Even where EDAC has general application (e.g., in memory), the relationship between EDAC and fault monitoring for higher levels of system reconfiguration or maintenance in place is uncertain and is approached on a case-by-case basis if at all.

TECHNICAL ISSUES

In this section, the technical issues that must be addressed to bring the technology of active reliability engineering to maturity are listed and described.

Fragmentation of Efforts. Efforts to increase reliability of systems must originate within several disciplines of engineering theory and practice.

These include system architecture, software engineering, testing and design verification, design of database management systems, computer networks and communication systems, component and packaging engineering, field operation and maintenance, and others. Although they all have a common goal, these efforts have remained largely disjointed. A definite lack of common viewpoint and of systematic communications is evident at the present time. There is also a gap between the results of theoretical investigations and practical engineering solutions to fault tolerance problems.

Lack of Cost/Benefit Measures. Thus far there are no general methods for a convenient quantitative assessment of the benefits (in terms of life-cycle cost reduction) of active reliability. The initial extra costs due to the various redundancy techniques are more directly evident and tend to bias a large class of users (who do not have an absolute requirement) in favor of systems without active reliability measures.

Lack of Specification and Acceptance Tests. The user community at large has inadequate knowledge of the properties and limitations of fault tolerance. As a consequence, specifications for reliability do not address active reliability. Active reliability cannot at present be subjected to the same level of acceptance testing as other parameters. For example, reliability requirements for a given time interval do not specify classes of faults and do not define what constitutes acceptable recovery. Also, the MTBF specifications do not explicitly deal with fault classes (e.g., transients and design faults) and recovery requirements, and they ignore the differences between redundant and nonredundant designs. Extremely high reliability and MTBF predictions are sometimes offered without stating the implicit assumptions of a static reliability model, and they address a very limited class of faults. In contrast, speed requirements in instructions per second can be stated and tested for acceptance precisely.

Institutionalization. Introduction of active reliability in system design requires early commitment in the evolutionary design of the system. This in turn requires that the techniques and concepts become well established in the design community. While the number of active reliability techniques to serve as maintenance aids to commercial equipment and systems (especially computer systems) has been increasing, none of the major manufacturers has announced a fully fault tolerant line of computers or subsystems for military requirements. The only fault tolerant systems that have been actually delivered were custom made to special requirements. There is significant commercial interest however. The military must provide the impetus to foster wide use of active reliability in its systems.

Validation and Verification. A significant problem is the need for validation and verification procedures from the systems acquisition manager's perspective, beginning with operational requirements and proceeding through systems acceptance test and evaluation. Without verification procedures it will be difficult, if not impossible, for the Government to put an active reliability requirement into a system specification. There must be a clearly defined procedure for the Government to use in purchasing a system from a contractor; otherwise the active reliability specifications will be only design goals rather than design requirements.

Estimation of Confidence Limits Testing For Large Logic Networks. Another problem is to accurately determine confidence limits for the reliability of large digital logic networks (e.g., VHSIC chips), without exhaustively exercising all possible input sequences or simulating all logical faults in the network. This project must address development, implementation, and demonstration of a logic simulation and fault analysis system to serve as a tool for determination of reliability confidence limits in digital networks containing up to 45 000 elements.

Functional Test Design Theory. Functional testing is required, from its theoretical foundations to principles of application in systems, software, and hardware. This program can become the basis of functional-testing microcomputers as well as analog functional testing. Functional testing has often been omitted in system design until the last stages. This procrastination leads to testing as an add-on, rather than a built-in, function. If a good applied theory of testing is developed, functional testing can become part of the basic design.

PROGRAM APPROACH

In previous sections, the concept of active reliability was defined and the role of active reliability engineering was developed. Technical issues needing resolution were identified. Here the approach to be taken by this project is outlined. Many technology development programs are structured to develop their technology in a vacuum, i.e., apart from real-world system projects and requirements. Their object is to derive a "generic" technology base to be applied to problems only after it is fully developed. Unfortunately, this type of approach takes far too much time prior to the realization of any tangible benefit. It also requires an excessive up-front research and development investment prior to application.

In the program proposed herein, on the other hand, the technology will be developed by means of a series of incremental steps. Each step is directed at solving a real application-related problem while it furthers the development of the technology in general. These steps are not directed at solving the problems of specific programs or system developments. Rather, each will address active reliability engineering issues for a general class of system applications. Each one of these classes addresses a real system problem the Navy is faced with today. In addition, there is a step-by-step progression of increasing scope and universality. Thus as the steps are completed (i.e., as the active reliability problems for system applications are solved), the technology of active reliability engineering is brought to a greater level of maturity and generality.

The advantage of this approach is that it is directed to the solution of real problems as the concept of active reliability is developed. The technology is immediately applicable. A separate program to apply it is not required. Many generic programs have difficulty in finding their direction upon initiation. Considerable time is spent in defining requirements and goals to strive for. This is not a problem for this program because each step provides its own direction both in terms of initial requirements and initial directions to be taken (problem to be solved). Although three separate steps are envisioned and some time phasing between the three is anticipated, they are not strictly sequential. In fact, all three should be started at about

the same time. More effort should be placed on the first step first so that its outputs, both interim and final, can be utilized by the other two. The approximate time phasing of the program is shown in figure 1.

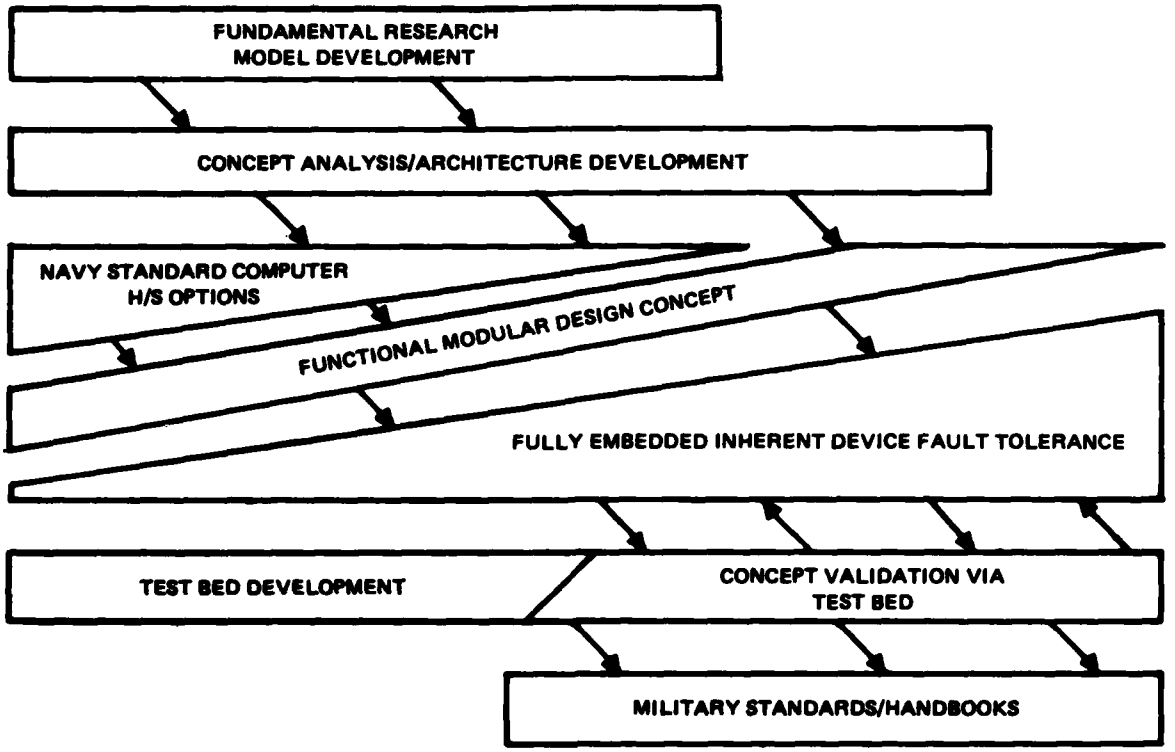


Figure 1. Active Reliability Engineering Program activity phasing.

The first step is directed at the problem of active reliability for networks consisting of Navy standard embedded computers. It is anticipated that many system designs will utilize networks or arrays of several to 50 Navy standard computers such as the AN/UYK-43 or AN/UYK-44 as their computing resource. These networks present a problem and an opportunity for active reliability. The problem exists because these networks will be sensitive to a failure in any one of the many computers, and the reliability of the individual computers will not be great enough to prevent their occasional failure. The opportunity to apply active reliability exists because the distributed nature of the network is ideal for application of active reliability engineering procedures on a network basis. In the progress of the step, hardware units will be defined and designed. These units are intended to be purchase options for procurement in the UYK-44 computers that are to be placed in network configurations. These hardware options, which will be totally integrated into the computer design, will be used to render the network actively reliable to the degree required by the network application. As this development takes place, more or less general principles for active reliability engineering will fall out, especially those applicable to functionally distributed systems. A detailed description of this step is provided in Section 3.

The second step applies to design by functional modules. Many systems consisting of one or several cabinets could be designed to a large extent to use predefined functional modules. Each module is designed to perform a given

function and is similar to all others of identical function. This level of design addresses systems smaller than those of the first step but is applicable to a broad range of signal processing and control applications. The goal in this step is to define subfunctions to be included in each functional module to render the overall system actively reliable. Again general engineering principles for active reliability are expected to come out. This step is described in Section 4.

The final step addresses the design of an individual circuit card by impacting the design of integrated circuit components. The main impact is to be on VLSI and VHSIC chips that will be used to implement systems on single circuit boards. This step must address very broadly applicable design concepts to be applied to all chips or all subfunctions on chips to make the chips and the circuit boards actively reliable. The goal is to achieve such a large degree of universality that these functions and design features can be incorporated as standard features of all automated design tools (CAD/CAM) for all circuit, equipment, and system design. This step is described in Section 5.

Although the three steps constitute the main development thrust of the program, there are several other areas of work that must be included. One is the development of general analysis tools and procedures. Although the main thrust of this proposed program is applied rather than generic, universal analysis tools, models, and methodologies must be included. This avoids a complementary problem associated with many programs that are too narrowly directed to have any application to other than a single system. In this program there is coequal emphasis of the generic analysis side of the technology along with applied development. This work occupies both ends of the development process. At the front end, the technology of active reliability analysis will be developed to include active reliability models, reliability analysis procedures, and validation and verification procedures. At the back end, as the three various steps wrap up, the universal application aspects of the technology they develop will be institutionalized in military handbooks, standards, and specifications.

A key feature of this program is a test bed to be used to build prototypes that exercise and validate the concepts being derived by the program. The role of the test bed is crucial to program success. Experimentation via the test bed will take place by means of modeling real-world system applications in hardware. In addition, software simulation and modeling will be utilized where advantageous. The test bed is a large-expense item but is very necessary to keep the program technically on track.

A detailed work breakdown structure is presented in Section 6. This program plan reflects the comments previously made and describes the activities that must be performed to bring this technology to fruition. Table 1 provides a funding summary based on these task descriptions. This funding picture includes the total estimated cost based on a 5-year duration. A year-by-year breakout over the 5-year period is provided and assumes FY 84 start and unescalated costs. Finally a breakout between funding categories is shown.

Total Cost

$24.9M

5-Year Plan

FY 1	FY 2	FY 3	FY 4	FY 5
$2.5M	$4.5M	$6.8M	$6.2M	$4.9M

Funding Categories

6.1	6.2	6.3a	6.3-6.4
$0.6M	$4.8M	$12.6M	$6.9M

Table 1. Proposed program cost data.

3 SURVIVABLE UYK-44 NETWORK OPTIONS CONCEPT

PROBLEM STATEMENT

A major acquisition effort to place a new generation of standard computers into the Navy inventory is underway. This acquisition addresses two new computers, nomenclatured AN/UYK-43 and AN/UYK-44. The UYK-43 emulates the instruction set of the UYK-7 computer and will replace it as the primary large tactical computer of choice in system designs. The UYK-44 will emulate and replace the UYK-20 and AYK-14 minicomputers. Both the UYK-43 and -44 will offer significant performance and other advantages over their predecessors.

One of the main advantages over the UYK-20 offered by the UYK-44 is much greater flexibility. The -20 is available in only one general configuration; the only option is the number and type of MIL-STD-1397 interface ports desired. The -44 will be available in two different processor speed configurations: 1/2 and 1 times the speed of the UYK-20. The -44 also will be available for purchase as a circuit card set or in a box. The card set will be capable of direct integration by the system designer into the equipment cabinet and cabinet circuitry. The box will be a stand-alone computer, with memory and power supply included and with interfaces to other separate cabinets via MIL-STD-1397 I/O channels. This is the only way the -20 is offered today. Further flexibility is offered in the ability to address more memory and memory of different speeds and to provide more I/O channels than the UYK-20 provides.

Additional advantages of the UYK-44 are improved reliability and maintainability features. The -44 offers increased reliability in the form of increased mean time between failure (MTBF). The militarized reconfigurable processor (MRP) version (the card set) is specified at 10 000 hours for this figure. The militarized reconfigurable computer (MRC) is specified at 5000 hours. Both figures exceed UYK-20 performance. The -44 will provide improved maintainability through hardware built-in test (BIT) and firmware diagnostic functions. The BIT will detect at least 98% of all MRP/MRC malfunctions not requiring external signals. The BIT and diagnostics will isolate 80% of all detected malfunctions to a single replaceable module, 95% to a group of two modules, and 98% to a group of three modules. Standard electronic module (SEM) standard formats are used as the form fit and function base for the circuit card modules.

The power and flexibility potentially available in the UYK-44 make it attractive for a wide range of military applications requiring computing resources. Of interest are those applications for which a number of -44s are combined in a network configuration to solve a large-scale tactical computing problem. One example of this class of application is shipboard or submarine combat system computing. Networks of -44 type computers have been proposed as the computing resource for several classes of combat systems as a method of replacing the single large central computer. Preliminary designs have been proposed for highly automated combat systems consisting of 25 to 50 computers for destroyer-cruiser types and about 40 for submarines. Smaller networks have been proposed as the implementation of control and data manipulation capabilities for major subsystems of the combat system. For example, a configuration of eight computers has been proposed for use in surface ship sonar.

A network consists of a number of computers (i.e., UYK-44s) each containing or connected to specialized hardware that interfaces the computer to a common data transfer mechanism. The combination of a computer and interfacing hardware is called a computer site or network node. The common data transfer mechanism is called a data bus. What makes a network just that is the fact that interactive processing by the computers is required to complete the network data processing tasks. The computers do not simply process data autonomously.

The network approach is considered feasible by the technical community responsible for the development of these systems. In fact, it is considered advantageous. There are technical problems to be overcome in implementing them, however. Perhaps not an inhibiting problem but a very real one is that of network reliability. This problem has two causes: (1) many computers are combined to form the network, and (2) the loss of any one computer could render the entire network inoperable. The impact of the first is that the reliability of an aggregate of small computers is lower than that of a single large computer if a fault in any one computer is critical. A single UYK-44 is much more reliable than a single UYK-7 (or a -20 for that matter). The problem lies in that one out of N UYK-44s will fault more often than a single UYK-7 if N is more than a few.

As for the second cause, any single computer fault in a network can be potentially critical. A single computer failure could render an entire network inoperable (or functionally useless) for several reasons. One is the tendency for an input-output failure in one computer to propagate through the entire network via the data transfer mechanism. Another is the loss of critical functions in the failed computer. A third is the loss of connection to external devices caused by the failure in the computer that attaches those devices to the network. Not all functions in a network, especially a tactical computing system, are critical at any given time, nor is every external device critical. But every computer is critical even if its function or device is only occasionally critical, because it is unknown which computer will fail at any given moment. The definition of critical boils down to those functions that could be called on before the computer resource can be repaired and the system restored. If functions and devices are tied to computers, then the loss of any computer must be assumed to be catastrophic to the network. A leading approach to this problem is to incorporate the ability to relocate functions and connections from computer to computer. More will be said about this technique later. It is important to note that to make any network resistant to faults within its computers, functions and device connections must be relocatable and propagation of I/O errors prevented.

The exemplary applications previously stated show significant differences in the magnitude of their requirements as reflected in the number and relative dispersion of computing resources in the network. A size difference spanning the range of eight to fifty computers was cited. The term "dispersion" addresses the physical and logical separation between the computers. In a small application, the computers may be interconnected by means of a local shared data bus such as the cabinet backplane. Indeed, the computers themselves might be MRPs vice MRCs, and interfaced at the card level. Larger applications will tend to require more separation than the confines of a single cabinet and thus to require more formally defined interfaces. These applications will more likely utilize MRCs as opposed to MRPs. Greater

physical separation requires that interface procedures become more formal and thus more complex (intercabinet vs intracabinet). The approach proposed as the solution to the network reliability problem must address this diverse range of applications.

Restating the problem, how do you achieve highly reliable computer networks though using many moderately reliable computers? Or stated more generally, how do you build networks to meet varying reliability properties—moderate to very high—with a few to very many moderately reliable computers? There is one further constraint on this problem, having to do with the computers themselves. The UYK-44 computers are being designed for single-computer applications as well as networks. They are being designed primarily for the simplex application. The basic design of the -44 does not or cannot provide unique features that support network reliability. These features will have to be added if and when their development is undertaken. In addition, the -44 is designed to accommodate interface standards that have been created with the single computer in mind. The standards were not designed to support computer networks or an active reliability philosophy that assumes that the computer side of the interface can fail. Restating the original problem, how do you build a reliable network with UYK-44s designed to meet existing interface standards?

In the next three sections, seven techniques for solving this problem will be advanced and briefly analyzed. They fall into two categories. The first four techniques address the problem at the level of the computer itself. They are all based on the concept of making the network more reliable by making the computer more reliable. The remaining three techniques are based on the concept of making the network as a whole more reliable without necessarily making each individual computer more reliable (i.e., networkwide active reliability engineering). They deal with the network as a system and the computers as subelements.

FOUR TECHNIQUES BASED ON UYK-44 FAULT TOLERANCE

This section describes various techniques that would simply make the UYK-44 computers more reliable. For example, an increase of an order of magnitude in the reliability of the computers would, in most cases, render network reliability as good as if not better than a large central computer. There are two general methods of achieving improved reliability in these computers by active means. One is the application of brute force fault tolerance and the other is reengineering the -44 to be highly reliable by using active reliability techniques. Descriptions of three brute force techniques are followed by a general description of the reengineered approach.

Brute Force MRC Replication

The most basic brute force approach consists of triplicating the entire -44 computer and adding voting circuitry to make a highly fault tolerant unit at each computer site. Thus every single small UYK-44 in the network is replaced by a rather large but fault tolerant one. Even with triple redundancy, the reliability at each site will probably not be good enough to support a 60-day mission. Highly fault tolerant translates into highly reliable only with some qualification concerning mission duration. Some modifications could be added to allow maintenance in place on the failed one out of the three while the other two continue to run, or the triplication

scheme would have to be modified to allow selection of a hot spare. The brute force approach is technically feasible in a reasonably near term.

The brute force approach is feasible but might not be considered practical. Full triplication with voting would imply that the network hardware costs would rise considerably. Addition of a hot spare makes it even worse. An acquisition manager could realistically face a quadrupling (or more) of his hardware cost. In terms of dollars, this implies an acquisition cost from $2.5M to over $10M per network installation. This is a considerable cost problem to be overcome but it is not the only one. All computers would have to be thoroughly tested between missions to ensure that no long-term degradation was happening. The failure repair rate (number of units in the repair and logistics pipelines) would at least triple, causing a large increase in the cost of ownership. Triplication with a hot spare increases the problems mentioned by 25%. In addition, intermittent and power-up faults could render this approach useless by prematurely forcing spares to be committed on passing faults. These approaches may be practical from the perspective of a very reliable single central computer, where the hardware multiplication factor is small, but may not be practical for a network.

Brute Force MRP Replication

A less comprehensive brute force approach would be applied at the MRP level rather than at the MRC. The previous example called for triplicating entire computers; this one proposes triplicating the central processor only. Memory units would be triplicated only on a selective basis. For all but the most time-critical functions, memory would be replicated in a less extensive manner, most likely on a module-by-module basis. A given memory module (and the functions it contains) could be backed up by one module, two modules, or less than one module (shared), as required by the application. The reliable (triplicated) MRP would contain software to monitor memory failure (assisted by memory error detecting and correcting hardware) and to control the memory switching process.

This approach has several advantages over the previous technique. It does allow some tailoring of each computer node (site) in the network to requirements for reliability based on the criticality of that node. It also allows maintenance in place on the memory modules. Its principal advantage is the lower cost magnification required to achieve replication. The cost would be increased by a factor of two to three rather than three to four as for the previous case. The problem with the repair pipeline would be reduced somewhat. This approach suffers from a number of disadvantages. The hardware cost factor is still quite large. Specialized hardware and software would have to be developed. The software to control the switching process could become fairly complex, especially if minimization of hardware cost were emphasized. Since the processor itself is triplicated, all of the problems identified with the first approach would also apply at a somewhat reduced scale. This approach is considered feasible, but some development is required. Fault tolerant research into this approach is underway for the -44 based on a NASA-initiated concept of this type.

Dual Redundant UYK-44

Another possible technique is to duplicate each UYK-44 at the MRC or MRP level. With two computers at each site, voting would become impossible because the voter would not know which was correct and which was in error. To make this approach work, the voter is replaced by an extensive fault monitoring and error checking unit. This capability could reside in a separate stand-alone circuit card or could be incorporated into the UYK-44 circuit cards. A stand-alone card would be a less cost-effective approach but would require considerably less development time and effort. At least one UYK-44 is forced to run off-line but accepts inputs from the network; its outputs are held off-line (disconnected). If the other UYK-44 registers an error, then the error checker takes it off-line and connects the second to the network. The errant computer can then be taken out of service for maintenance. Both computers run identical software. The error checker need monitor only one computer (which will be the primary computer), but to monitor both would be beneficial. The error checker must ensure that both computers are synchronized. For the MRC-level application, the computers are duplicated in their entirety. For the MRP-level application, only processors are duplicated methodically. In this case, memory is replicated on a case-by-case basis, as discussed for MRP triplication. This will require additional software in the MRP to control the reconfiguration of memory.

The main advantage of this approach over the previous two is that it would achieve essentially the same result but at lower hardware cost. A hardware purchase cost increase factor of 1.5 to 2.5 can be anticipated. The actual figure would depend on the amount of replication of memory desired. The lower figure represents MRP duplication and minimal memory backup. The higher figure represents full MRC duplication. In addition to two computers or processors, an error checker would be required at each site, causing more than a doubling of the actual hardware needed. The error checker would be functionally complex, but could be implemented in several VHSIC-level chips and therefore would be relatively inexpensive. It would require development, thus the expenditure of R&D money, and could cause a schedule risk for near-term requirements. The lower hardware cost approaches would require more software to control the memory reconfiguration efficiently.

UYK-44 Reengineering

The final technique for making all the computers of the network highly reliable and thus making the network reliable is to reengineer the -44 itself. By applying active reliability techniques, a highly reliable MRP could be developed without resorting to complete triplication. Memory would be treated as discussed in the previous approach, or error detecting and correcting codes could be used. I/O channels would be replicated or redesigned for high reliability. The main problem with this approach is the cost and time required to redesign the processor unit. The actual cost and time factors to be encountered will depend on the detailed design of the UYK-44 selected as a result of the competition between the two contractors. Five years would be a not-unusual timeframe. It is important to note that within 7 years, more reliable -44s potentially could be available by virtue of incorporation of VHSIC technology and its higher inherent passive reliability due to its larger scale of circuit integration. The active reliability approach would result in a smaller cost increase than the previous two techniques—approximately 1.5 to

2.5—once the redesigned processor version is available for procurement. Although no known funded work is underway in this area, some ongoing related industry IRAD work is known to exist and could have potential application.

Computer Reliability Enhancement Summation

Four techniques for attaining a UYK-44 fault tolerance solution to network reliability have been presented. All of these techniques will make the UYK-44 more reliable at varying costs and degrees of near-term feasibility. In addition to the individual problems cited for each technique, they all share some additional common problems. In the main, the considerable expense required in hardware provides tolerance of no more than a single fault, and this additional hardware may not tolerate faults in the voter circuitry unless the hot spare concept is used, which would further escalate costs. There is no provision for adjusting this technique to provide even more tolerance for critical points on the network except within one processor. That is to say, this technique cannot provide greater depth of coverage for immediately critical functions at the expense of routine or potentially critical functions across the entire network. This is not a limiting factor if maintenance in place can be provided but does lead to reduced reliability effectiveness compared to other approaches. It is important to note that none of these concepts directly addresses the how of maintenance in place, i.e., the mechanics of how it is to be accomplished.

Another problem is that backup and replication of functions occur locally. A small computer in one location and the functions contained therein are replaced by a large fault tolerant computer at that same location. Functions are not dispersed since they are replicated within the large computer. This provides no way of tolerating problems that would massively impact one computer site while leaving the rest of the network alone. A good example is minor battle damage. Others are local electrical power and cooling disruptions. Again, a highly critical function is lost if its computer site is destroyed.

A final common problem involves I/O channel connections to external devices. Either the I/O channel circuitry in the computer is replicated, with voting circuits arbitrating the output to a single I/O cable, or entire channels are replicated. The former option presents the problem of single fault intolerance for voter or cable failures. The latter necessitates duplexing in the external device; thus, in many cases, engineering modifications to the devices will be required.

In summary the fault tolerant computer approach to network reliability is feasible although some development is required by all three techniques. It will be expensive in terms of incremental hardware cost increase and it won't be totally effective. Its main limitation is caused by its inflexibility to adjust to varying requirements or actual faults. This is because each computer site is treated as an independent entity for fault tolerant/reliability purposes. No advantage is taken of the fact that all computers are tied to the network in cooperative manner for other (operational) functional purposes. The lowest cost approach will require the most development and may not be feasible in a sufficiently near term.

THREE TECHNIQUES BASED ON NETWORK ACTIVE RELIABILITY

The next generalized approach is to utilize the resources of the network as a whole to support network reliability. This concept approaches the network as a system and the computers as components, then applies active reliability to the overall network. UYK-44 modifications are proposed that support the overall network active reliability concept rather than the fault tolerance of each and every -44. Thus any given computer node on the network is not fault tolerant by itself, but the overall network achieves sufficient reliability to accomplish its mission. There are several techniques that can be used to implement this concept, the variation being largely the degree of emphasis given to software vice hardware.

Software Implemented Fault Tolerance

The purest software approach follows work funded by NASA entitled Software Implemented Fault Tolerance (SIFT). This concept represents a blending of traditional fault tolerance with the network concept of operation. In this approach, functions are triplicated but not in the same computer. Each function would be performed simultaneously by three different computers. The output data resulting from each function or subroutine would be transmitted via the data bus to the next triplicated function and so on. A voting on the three inputs takes place in the software of the succeeding function. If a fault occurs, its effect will be ignored because it will have been performed correctly at two other computer sites. That's the theory. In fact there are several complicating factors. Additional measures must be taken to prevent the infamous "babbling node" problem. A babbling node doesn't know when to shut up. A fault in the network interface circuitry or a software error (caused by hardware faults or software) causes it to continue transmitting, denying access to the data bus by other nodes.

The babbling node "hogs" the bus and can disrupt the entire network. The interface circuitry problem is prevented by use of hardware fault tolerance techniques or triplicating the data bus. The software problem is more subtle because a software fault can cause the computer to try to gain access to the data bus repeatedly in an apparently normal manner. This problem can be solved by rapid detection of the most likely software-related problems by a highly reliable watchdog function that would look for a single computer that is generating output not corresponding to that of any two others. The errant computer would then be shut down by hardware means. Specialized software would have to be written for every node to handle the three inputs, evaluate them, and pass the correct functional output to the internal functions of the computer.

This approach provides somewhat more flexibility than the brute force techniques. For example, a two-correct-out-of-four approach could be taken for highly critical functions. Alternatively, several computers could be placed on the network in an idle mode for use as hot spares for any computer in the network, thus providing multiple failure protection for all computers. This approach requires a fairly rapid detection of a failed computer and the ability to switch nodes on and off. It would require hardware modifications to the -44. But it has several other advantages. It is an inherently software approach that can be used to tolerate purely software faults (i.e., errors in software) by using a different source for the code of each of the three replicated functions. Thus an error in software would be detected and

handled just as an error in hardware. It also requires a minimum of modifications to the -44 computers themselves.

The approach does possess some disadvantages. One is considerable hardware cost expansion. Again, this approach involves tripling to quadrupling the hardware required. Not only must each function be triplicated in CPU and memory hardware, additional hardware is required for the additional control functions and watchdog functions. Spare computers may be required. What may turn out to be a real limitation is the need to increase the speed of the data bus by a factor in excess of three: in addition to three times the outputs, some "slack" time must be provided for synchronizing the three outputs, especially if different software is written for each of the three copies. Additional data bus bandwidth must be devoted to control and test signals. This places a real technical risk on this approach; the timing problems and issues for a real-time combat system network have not been completely resolved even at basic capacity, much less at times three.

Although this approach requires little hardware development, it requires a considerable amount of software development. The executive program of every computer would require special routines to handle three inputs. Every possible way in which a problem could show up would have to be provided for. A network monitoring and control protocol and related software would have to be developed. Since software seems to deteriorate over time, various means would be required to verify the cause of a detected failure in a computer. If this is not provided, then progressively more computers will be "sidelined" due to software glitches rather than hardware faults. An extensive testing procedure would be required and would be run continually to replace these computers into service. This approach does not address the problem of the interface to external devices unless every such device is modified to triple-plex its input/output channel to the network computer. Finally, there is some real technical risk associated with the ability of this concept to work in a real-time processing environment of the magnitude of a combat system.

All but one of the techniques discussed up to this point are based on replication and voting, i.e., common fault tolerant practice. Triplication of hardware is required whenever the technique is applied. To avoid the problem of so much excess hardware, an alternative approach is required in which failures are detected and the system is reconfigured accordingly so as to maintain correct operation. In terms of a data processing network, this implies relocating or, more currently, reassigning functions from a failed computer to a working unit and shutting off the failed unit. Two possible approaches to doing this in the network are discussed next. The first is based on the proposition of making no hardware modifications to the UYK-44 computer itself, the second on making such hardware modifications.

Node Interface Unit Enhancement

In this approach, the UYK-44s on the network do operational data processing oblivious in their software design to the network or its active reliability considerations. The functional software required for network operation and network active reliability will reside in the hardware that interfaces the -44 to the network. This means that this hardware will contain another computer. Hopefully, this computer will be less complex than a -44;

otherwise we would have two -44s at every node. It should be less complex because it should be a slower device, i.e., it most likely would be a microcomputer. The network control functions should be less complex than the operational functions. Thus the node interface hardware will be built around a microprocessor.

The node interface unit provides functions to maintain control of the network as a whole. In doing this, it monitors the performance of its -44 (or -44s) for any evidence of incorrect operation. It also monitors itself and the data bus for any evidence that another node is not working properly. Upon detection of a fault, either it switches in another -44 if it has a local spare or it notifies the network to activate another node. For time-critical functions, the spare may already be loaded with software identical to that contained in the faulty processor. In other cases, the spare unit will have to be loaded with the correct functions. In any case, the spare computer is initialized, address tables in the node interfaces are updated to reflect the reassignment, and the spare is off and running. Finally, the node interface shuts itself or the faulty computer off. In some cases a node interface unit (NIU) will not detect errors in its computer or within itself. These errors will be detected by other NIUs that must vote on the performance of the errant node. Once the vote goes against the node, it must be shut off. Some latency is encountered in this approach.

This approach offers more flexibility than previous ones since the network interfaces and controls can be programmed for varying degrees of fault tolerance—one, two, three, or more faults per function. It also allows maintenance in place and modification around a fault via the network. In fact, the node can go off-line and run a diagnostic program by itself once it is disconnected from the network. The additional cost required to provide this capability is less than previous examples, but difficult to estimate, since the NIU is there primarily to control the network interface and its ability to support network reliability is an additional function. Additional memory will be required in the NIUs to support the added functions. Obviously the spare nodes represent an additional cost factor. Therefore it is estimated that the hardware cost would increase by a factor of 1.5 to 2 over the basic network.

While the hardware cost might prove acceptable for this approach, the software costs might kill it. Although the software in the NIU would require only a microprocessor to run in, there could be a lot of it. There are several large software functions to be accommodated. One is the fault detecting software. Although the BIT and diagnostics in the -44 are moderately good, they are not adequate to support network reliability and they are not continuous. Since the rule for this option is no -44 modifications, external software (in the NIU) will be required. Another large software item will be the function-to-node assignment algorithms and tables. This software must control and modify all address tables that locate functions in processors and nodes in a dynamic manner. It must also keep track of initialization parameters and related state tables. Another software item will be the network control software or network executive. Its role is to maintain the network operating system. All nodes contribute to this executive, creating a total network operating system resident in the NIUs. In this case, the UYK-44s are treated as external devices and their errors are handled just as in the case of any centralized computer today. Thus, logically, the NIUs and

data bus become much like a large central CPU, and the -44s become peripheral processors. This approach defeats the whole intent behind the network and has no benefit other than creating job security for an operating system vendor. This network operating system could easily run into several hundred thousand words of memory per NIU. It would face years of use before being cleared of bugs and might never work in a real-time environment. In fact, this approach encounters one big technical risk factor: will the detection of faults and nework reconfiguration take place fast enough in the real-time processing world to be effective? At this point, nobody knows, but the risk is high. This approach does not directly address the problem of external devices. Another risk area is fault latency.

Hardware Options

The final approach discussed herein is similar to that previously discussed but allows hardware modifications to the UYK-44. "Modifications" is the wrong term; "additions" is more appropriate. Certain hardware elements can be added to support the phases of network active reliability. A hardware unit can be added to aid in detection of UYK-44 faults by various means. Another unit would shut the -44 off from the network data bus or external device. Another unit would be used to notify the network controller—now residing in a -44 rather than a collective of NIUs—to initiate reconfiguration. The modular design of the -44 supports this concept. These additional hardware units would be configured on SEM cards, as in the UYK-44 itself, and purchasable as options to a UYK-44 procurement. Thus there would be no need to design two kinds of -44s or to design a network-compatible -44 to be forced on all users. The hardware units mentioned are only exemplary; the exact definition and specification of the units will result from the proposed development effort.

One major advantage of this approach is flexibility over and above the one previously mentioned. A greater variety of active reliability approaches can be supported. At one node, for example, a single -44 can be duplicated, with the duplicate running in a hot spare mode. Its local switching in could be completely transparent to the network. This would make a single -44 node highly reliable. Three -44s could be put together at a site to achieve ultra-reliability. This does not preclude relocation of functions from site to site, even from sites made highly reliable as discussed above. However, the functions needed to manage this process become simpler and don't need to reside in the NIUs. They can reside in one or more -44s that are made highly reliable as discussed above. Thus the problem of creating a large-scale distributed operating system goes away. This approach would cost less, as little as 20% overhead in minimal cases. It would also allow human-in-the-loop active reliability at even lower costs.

The primary disadvantage is the technical risk that does remain with timing (latency and data bus overhead). The risk is less in this case, but does have to be resolved by research and development. In addition these hardware units have to be defined, prototyped, tested, specified, and placed into production. This is a schedule risk for near-term projects.

ANALYSIS

In summary, seven approaches to the problem of making a network of UYK-44 computers reliable were discussed. While all had some advantages and disadvantages, the hardware cost factor is one of the most significant in a comparative analysis. It is doubtful that any acquisition manager will be willing to pay more than twice the cost of the basic network (one without reliability features). This precludes the standard fault tolerant approaches, applied either to the computers themselves or to the network as a whole (SIFT). By this criterion, the only cost effective approaches are those that entail active reliability engineering. These all entail some risk factors, however, and research and development dollars must be spent to resolve the risks. The application of active reliability engineering to the computers themselves entails a risk mainly of schedule. The two applications of active reliability engineering to the network as a whole involve technical as well as schedule risks.

The latter two applications are fairly competitive. The advantage of the one lies in there being no need to develop new hardware other than the NIU itself. It has two major disadvantages: the technical risk associated with network timing (and the related question of the ability to effect timely recovery) and the schedule risk associated with the ability to deliver a large bug-free network operating system in a reasonable time frame. This application will require a unique operating system for each new network.

The advantages of the other are much greater flexibility and lower hardware costs. It can be adapted to a wide range of network sizes, even to a single -44 installation. Also, it can be adapted to a wide range of requirements and addresses the issues of maintenance in place and external devices. For these reasons the second approach is recommended for this program and for -44 networks in general.

APPLICATION

The concept proposed by this program is centered around a number of hardware and software purchase options to the UYK-44. The hardware options would be configured in SEM card formats compatible with the SEM form factor used for the existing -44 circuit cards. The software options would consist of application packages or modified standard computer executive programs. These options would be transparent to the non-fault-tolerant user. The purchase options will be configured in a variety of ways to build active reliability into a network or into a stand-alone UYK-44 to the degree required. Some of these options would entail considerable functional complexity (perhaps rivaling that of the basic -44 itself) but are considered economically feasible if based on very large scales of integration.

This proposal does not reflect sufficient analysis to provide specific definitions for the individual circuit cards. Their detailed definition is a primary goal of this program. However, some idea of the role that the cards would play in the network can be observed in their approximate definition. Proposed card descriptions are provided in the following paragraphs. It must be emphasized that these are preliminary functional definitions, not detailed descriptions of exact cards. Further analysis will certainly result in reorientation of some functions between cards.

A key card is an error checking card. It would play much the same role as the error checker mentioned for the dual-redundant fault tolerant approach. In this case, however, the error checker simply reports an error state to other functions, either to the network via a network interface card or to a local error handling card. The error checker works in conjunction with the UYK-44 BIT and firmware diagnostics and conducts its own tests to obtain a high degree of confidence in assessment of the correctness of -44 operation. There may be some latency in the detection of some classes of faults, but it is hoped that the relative number of these faults will be minimal. This is a subject for the analysis. The card would also contain extensive self-testing functions and test verification procedures. A fault record would be compiled on the card. Important to the flexibility required by this concept is the fact that the error detector card simply finds errors when they occur. Several other cards use this information. One is a local error handler card. This card would serve to shut the -44 down in an orderly manner. It could be used in conjunction with the error checker to implement dual-redundant -44 fault tolerance. Another card would be an autodiagnostic one, used to diagnose and locate faults to a card within the -44 when an error is detected. It would replace software diagnostic routines. The card would aid a maintenance-in-place philosophy or maintenance in general. In conjunction with the error handler, it could be used to achieve a self-repair capability in a UYK-44. This would effect an active-reliability-engineered capability for the individual -44. Finally, these error indications could be used to inform the network interface card.

The network would be informed via an active reliability network interface card or function placed in a standard interface function. The network would perform processing based on the detection of an error. As a result of this processing, control is exerted on individual -44 sites. A series of cards must be defined to allow this control. A network error handler card performs functions similar to the local error handler but under network control. The main functions of this card are to take the UYK-44 off the network in an orderly manner and to disconnect the external devices it is connected to. A specialized card is required to switch the I/O channels of a single -44 off the external devices. This card may replace the existing I/O cards, for network reliability applications. This card would be used if the external device has duplex channel capability. If it does not, the channel switching must occur at the node by means of another I/O switch card.

The network configuration under these switching capabilities would be changed in a dynamic manner. Therefore, the instantaneous logical configuration must be recorded somewhere to adjust the flow of control and data. This is the function of the network configuration control card. This card would contain the address and state tables that would keep track of the network configuration. These tables would be accessed as part of any network access procedure by any network computer, and then would operate under exclusive control of a master table contained in a network control site. Another card would be used to implement the network executive program segment in each UYK-44. A final card from this proposed set is an active reliability memory control and switch. This card would assist and control error detection and reconfiguration of memory modules. It would be an option available in addition to various levels of error detecting and correcting codes in the memory itself.

It is envisioned that the hardware options would be available for purchase as optional circuit cards and associated technical data explaining their use. In addition, various computers based on these modules would be available as complete units. Some of the possible computer types are described. As mentioned previously, the error checker, local error handler, and auto-diagnostic card could be used to achieve a self-repairing -44 at the MRP level. Combined with memory and I/O control cards, the entire MRC would be self-repairing. This would involve the cost of duplicating the internal circuit cards, which should prevent its widespread use in network applications, but it will find some roles (see below). In many cases it would prove more cost effective to use an error correcting option for memory as opposed to duplication. However, both options would be available.

The error checker, local error handler, and network interface would be used to build a network UYK-44 intended to contain noncritical functions and interfaces. This computer would inform the network and shut itself off upon detecting an error. It is the role of network software to determine what to do about restoring the functions. Replacing the local error handler with a network error handler would allow direct movement of functions to elsewhere on the network. This would be used primarily for a -44 that serves to interface external devices to the network and does not contain many other critical functions.

The error checker, network interface, network error handler, network configuration control, and network executive interface cards would be used for a network -44 that contains critical functions but does not interface devices to the network. These cards render the computer able to detect a broad class of its faults, report the fact to the network, respond to network commands rearranging functions, and adjust its configuration tables to respond to the changing network environment. Thus configured, this computer is suitable for switching itself and its contained functions in and out as a unit. The addition of a memory control card would give more flexibility to enable movement of functions on an individual basis. Addition of an autodiagnostic card would aid maintenance in place.

In any network, there would be a need for the network active reliability control functions to be located in a host that is highly reliable or ultra-reliable (even with the maintenance personnel taken into account). This need not be a problem for this approach. In the near-term, several levels of fault tolerance for one -44 could be built up to use the same optional circuit cards. The dual approach is one strong possibility. Another would be to combine the dual approach with self-repair on the primary unit of the two to achieve two-fault depth and maintenance in place. This would be an expensive option, but would affect only one site out of a possible 50. In the long term, a reengineered -44 (engineered for active reliability) is the best option for this site. Figure 2 illustrates a selection of potential UYK-44 units that could be configured from the optional card set. The basic UYK-44 card set is represented by the four boxes at the top. The next row shows six possible configurations each exhibiting a different level of support to an actively reliable network. The bottom row shows the set of seven common card types that would be used to modify the basic UYK-44.

The UYK-44 network problem is very real and must be addressed if these networks are to be reliable when put into use. It can be ignored for the time being of course. However, this only means that the problem will have to be addressed after units are in service, which is much more difficult. It is also a facet of a much more general problem. This proposed program is structured to address both the functional modular design concept and VLSI/VHSIC active reliability, discussed in Sections 4 and 5, respectively. The detailed work breakdown structure is found in Section 6.

RECOMMENDATION

The approach proposed herein is to investigate several system designs for which a network application is under consideration. These system applications will be used as a source of requirements from which a detailed network active reliability concept will be developed. As this concept is advanced, the functions to be required at the various -44 nodes will be defined on the basis of SEM circuit card formats and VLSI levels. Simulation models of the circuit cards, -44 computers, and network will be constructed and exercised. Finally, breadboard models will be constructed and tested in a test bed environment. Formal specifications will be generated after this concept and the resulting hardware units are validated.

AN/UYK-43 ACTIVE RELIABILITY

The AN/UYK-43 is being developed for replacement of the AN/UYK-7 computer in tactical computing applications requiring a single large centralized computer or a complex of such computers. To fulfill these applications, the -43 will contain fault tolerance features (actually active reliability, including maintenance in place). No new fault tolerance features are proposed for the time being.

However, the fault tolerance capabilities could be greatly extended in the future. What is suggested herein is for a significant increase in -43 active reliability at the point of a midlife technology upgrade. At this point the -43 can be modernized by replacement of the original circuit cards with redesigned cards that use the most advanced technology of the time. At that time, extensive active reliability features could be included to achieve a significant increase in reliability over and above that which will be obtained by the use of more advanced technology. It is proposed that the definition and specification of the optimal fault tolerant approach be undertaken now as part of this program so that it will be available as part of the contract package to be given to the contractor assigned the task of developing the midlife upgrade.

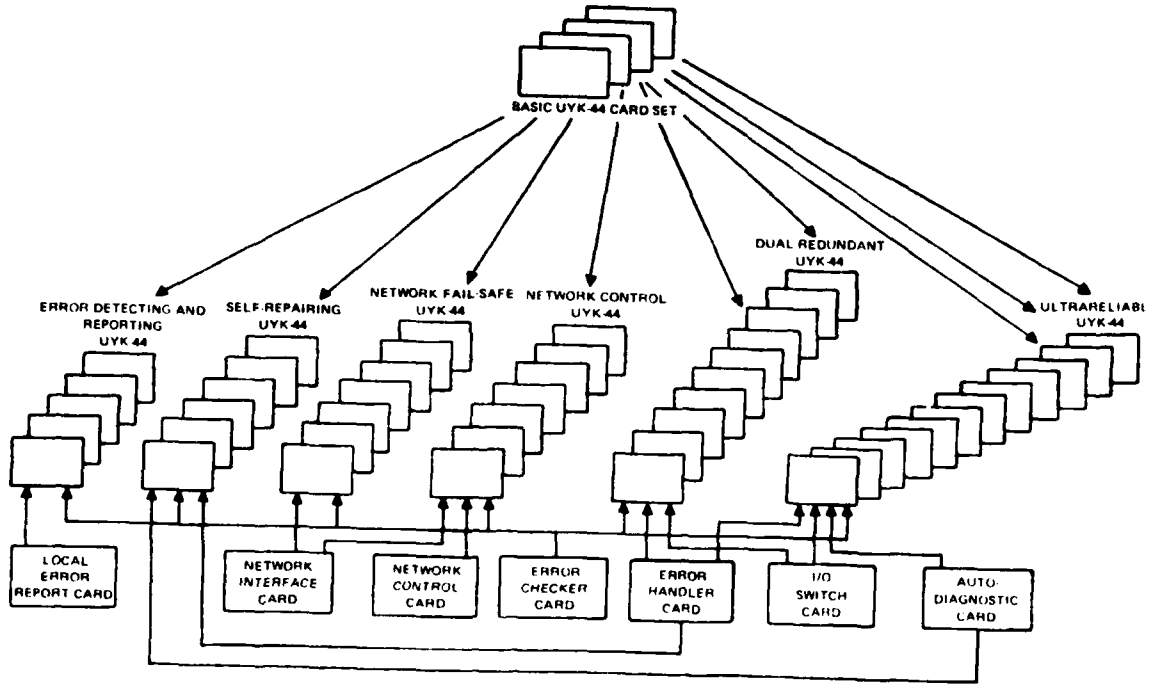


Figure 2. AN/UYK-44 hardware modification options

4 FUNCTIONAL MODULAR DESIGN CONCEPT

The problem of engineering active reliability into networks consisting of UYK-44 computers was considered in the previous section. An approach to the problem was advanced. It turns out that the network example can be used to establish a more general approach to another class of active reliability engineering problems. These problems are applicable to the cabinet and cabinet grouping level of system design. There are certain properties the network has that make the concept applicable. These same properties serve to distinguish the class of cabinet level problems that can be addressed by this type of approach. They also form a set of underlying assumptions inherent to the approach.

One assumption is the fact that the network concept is based on functional modularity of the system (i.e., the network) as a whole. The network is, by definition, broken down into pieces that are separate and distinct from each other, which is simply the definition of the term modular. Each node is defined and configured as a separate entity with respect to all other nodes once the functions it must perform to support the overall network are defined. There are some overall network configuration constraints, but the node is defined largely by itself.

The next assumption is the thesis that a limited set of functional elements to support active reliability can be defined as part of each module and that these elements possess some commonality from module to module. This does not imply that every module (i.e., node in the network example) has the exact same set of functions as every other module. Rather it implies that a core set of functions can be defined on a general basis. This set must provide all that is needed for active reliability techniques to satisfy all requirements at all levels. Furthermore, these functions must be modular themselves—they must be distinct from the system operational functions (in logical sense even if they share the same hardware).

The modules that make up the system must be compatible with the active reliability concept without necessarily implementing it. (It is implemented by the incorporation of the additional previously mentioned standardized functional elements into the modules.) However, the modules must support these functions. They must be controllable by the functions. They must be testable or monitorable by the functions. The UYK-44 computers are compatible by virtue of their BIT capability in assisting in fault detection and SEM module format of their circuit cards. The SEM cards containing the active reliability related functions are designed to be compatible with the computer's SEM format and interfaces.

There must be a common shared data transfer mechanism spanning the range of possible substitutions for a given module. This can be a data bus as in the network example, a cabinet backplane, or a shared data switch. The modules must interface to the mechanism in a similar manner. The mechanism must have the fan-out and addressing capacity for the additional redundant modules. The modules or one of the functions must contain some software or other "intelligence" (i.e., evaluation and decisionmaking) capabilities. There has to be control exerted over the error-detection and system-reconfiguration capabilities.

These statements provide a baseline definition of an equipment environment conducive to this type of approach. It is worthwhile to describe some generalized examples. The simplest to describe is a distributed or array signal processor. A highly complex high-speed signal processor can be built from a number of basic programmable signal processing units that could be configured to be identical or at least very similar (fig 3). With the exception of programming, they conform to the definition of the modules required to make the concept work. Since the signal processor consists of an array of these units, redundant units could be added to the array to facilitate operation with reduced data rate (bandwidth). Alternatively, full operation could be achieved via reconfiguration after the occurrence of a fault in one or more basic units. An array inherently contains the common data transfer mechanism required for the concept. What is needed is the definition of a sufficiently general basic unit by examination of several applications. Active reliability functions for the unit can then be defined. By using VLSI, these functions should require only a small part of the circuitry of the basic unit.

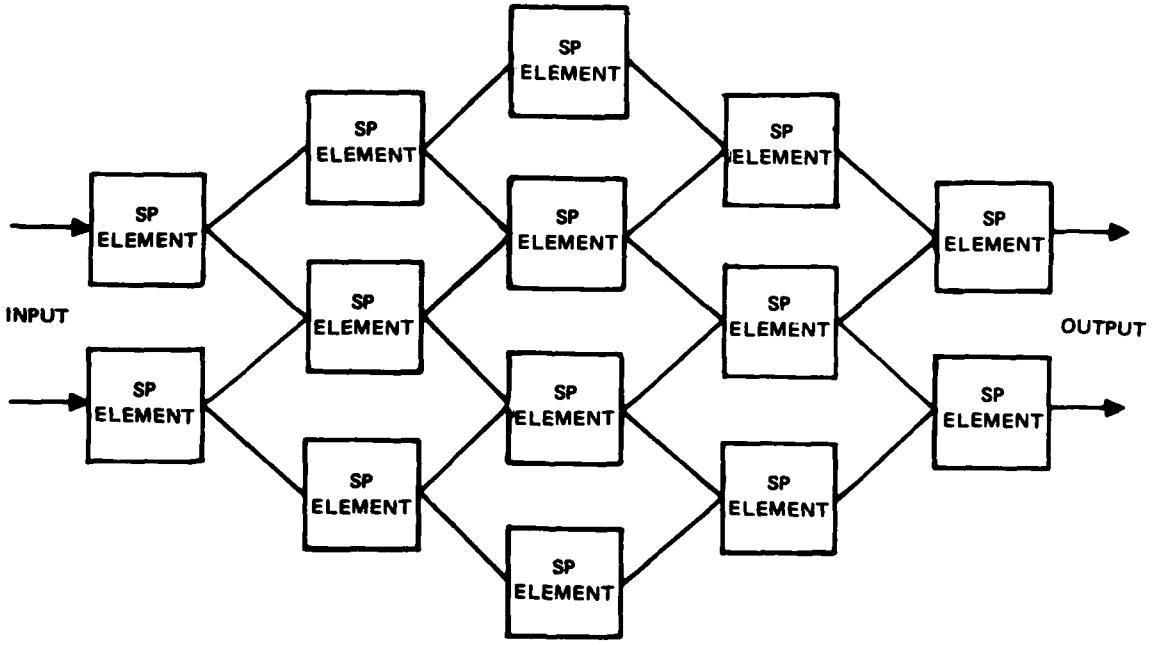


Figure 3. Distributed array signal processor application.

Another applicable design environment is distributed data processing within an equipment cabinet. In some applications, an equipment designer may choose to place one or more microcomputers within a cabinet to perform data processing functions. In many cases, it is desirable for these resources to be as reliable as possible. In these cases, a microcomputer data-bus-oriented approach compatible with many industry standards is usually taken. This is

directly conducive to the modular active reliability concept. There are two ways to go with this concept: one is based on identical computing units; the other is based on a set of microcomputer components interfaced to the data bus by a set of standard functional units.

For the first case, a single standard hardware module unit could be defined, from which a significant portion of the microcomputer equipment could be configured. For example the data processing functions required for a communications message processing system could be supplied by a number of standard microprocessor circuit boards (fig 4). Each would contain one or more microcomputers, an amount of memory, an internal data bus, and external I/O ports. A number of these would be configured into a system. Some boards would primarily play the role of memory, others primarily processing, others I/O. There would be no attempt to cram the maximum amount of capability into each board; i.e., this design approach would not result in a minimal hardware configuration. But each board would also contain the functions required to implement a systemwide active reliability concept, and this concept would be implemented, provided spare boards were added. The underutilization of each board and the added functional capabilities required would not have a major impact, assuming widespread use of VLSI technology.

For the second case, a standard microcomputer, standard memory unit, and standard I/O unit could be defined (fig 5). Each definition could include active reliability functions appropriate for that unit. For example, memory would be equipped with error detection and correction capability and controllability features. The standard CPU would contain extensive error detection as well as fault tolerant interfaces to the outside world so it could be shut off without disrupting the other units. The I/O unit would contain a combination of the features possessed by memory or the CPU. A multiple microcomputer of fairly large scale could be built with active reliability features. There is work ongoing on this approach. It has NASA funding for spacecraft applications (far different from weapon system requirements) and industry interest for commercial applications.

A third possible design environment is an extension of the second method described above for the microcomputer (fig 6). Consider a number of dissimilar functions that must be interconnected across a data bus. Although dissimilar, if they can be broken down into modular units, then a limited active reliability concept can be applied. Each functional module is interfaced to the data bus through a system interface and control unit. This unit monitors the module, controls its access to the data bus, and serves to shut down the module while notifying the greater system if it faults. Each module would be replicated locally to the degree required by its criticality. Noncritical functions would not be actively replicated. Highly critical functions would be given triple or quadruple redundancy.

It is proposed that this phase of the program investigate applications and propose initial solutions for all three design areas. This includes a rigorous examination of ongoing efforts by other agencies and industry to determine their applicability and to build upon their results. The concept will be developed through detailed design for one relevant application for each design example. At least one application will be modeled and validated in a test bed environment. The resulting concepts and techniques will be made available to various programs that use the design environments described.

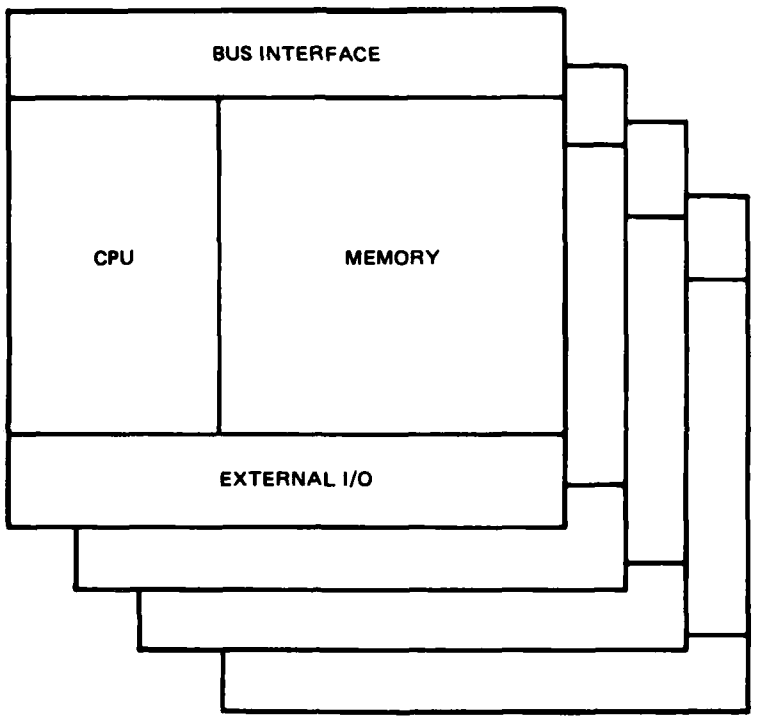


Figure 4. Standard microprocessor hardware board.

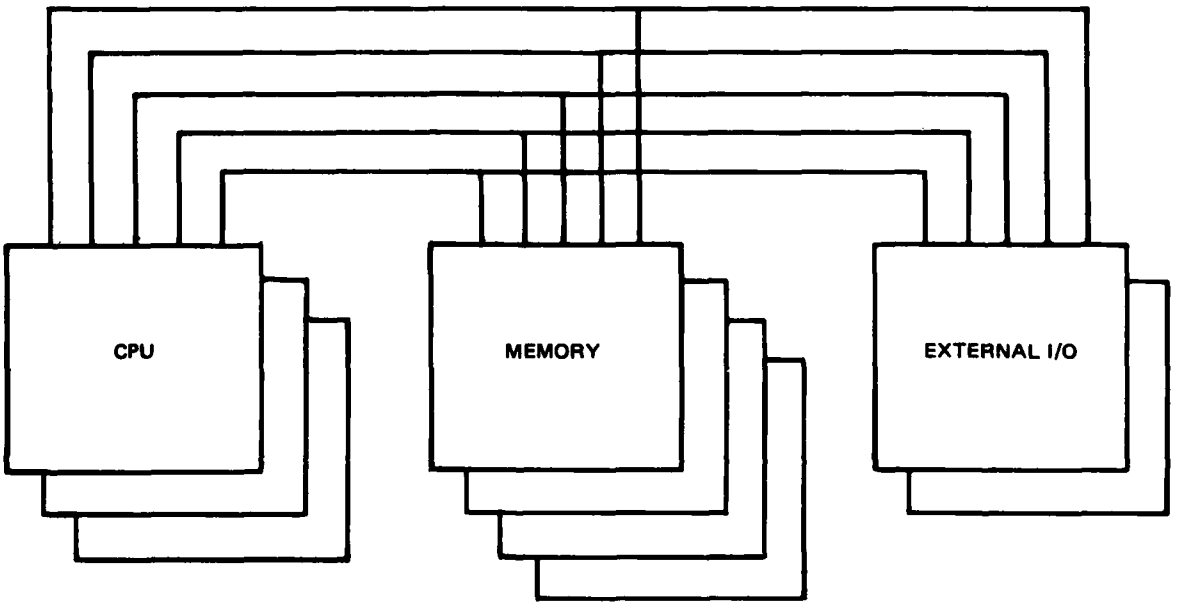


Figure 5. Standard microcomputing functional unit.

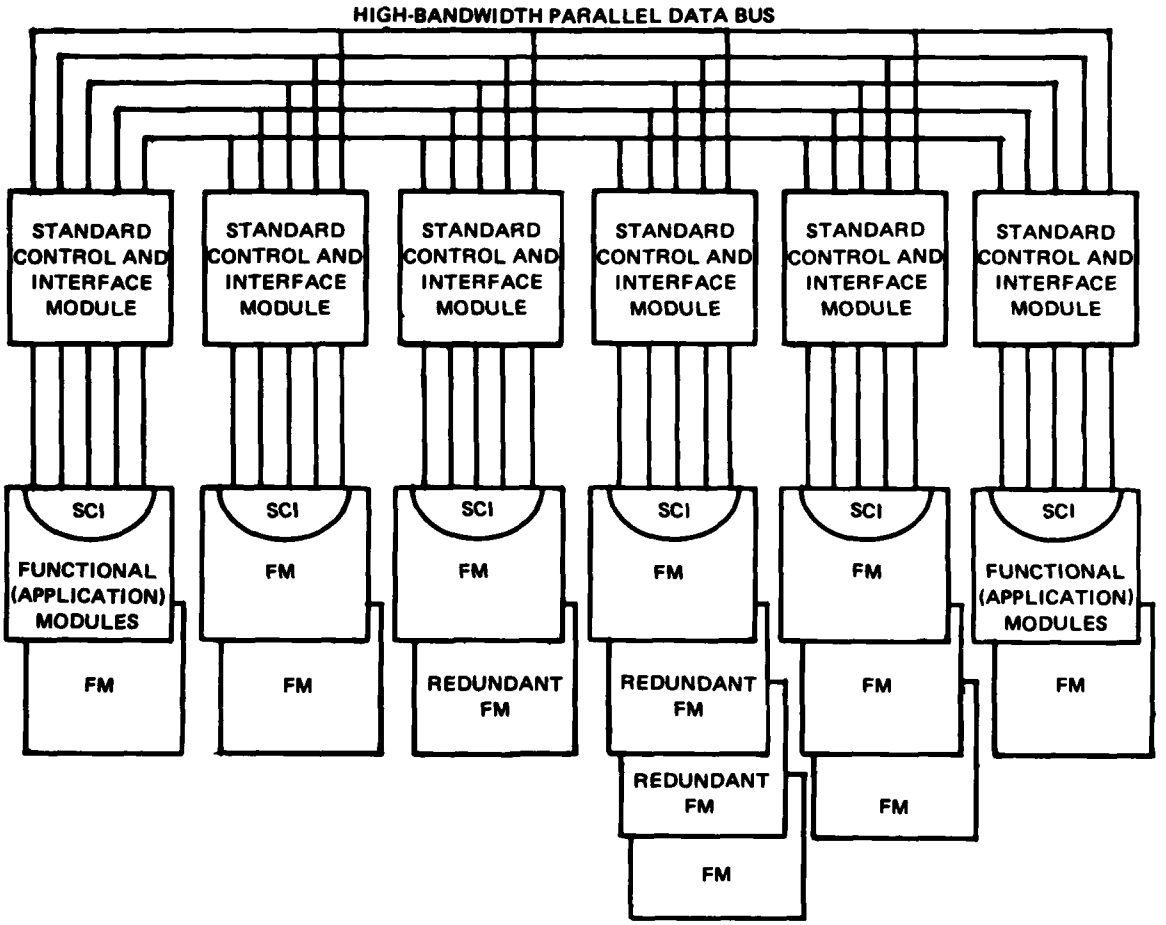


Figure 6. Generalized functional modularity.

The most important output of this phase of the program is a design philosophy or discipline for active reliability. As this phase proceeds, the design people involved with it will achieve an awareness of how to design circuit units to be compatible with active reliability. This awareness will consist of the design procedures or rules one will tend to apply to design if the question is continually asked, "How does this circuit fit into my active reliability concept?" These design rules will be simple and will not unduly constrain the design; and when applied from the ground up in a design, they will facilitate greatly an affordable active reliability concept. At the present time, this information will be gathered and placed into an active reliability design guide to be promulgated to the design community. Toward the end of this program, the design guide will be converted to a formal military handbook.

5 VLSI/VHSIC ACTIVE RELIABILITY

The emergence of very large scale integration and very high speed integrated circuits (VLSI/VHSIC) adds several new factors to the active reliability problem. The tremendous functional complexities obtainable on one or a handful of these chips make feasible many of the concepts presented here. These concepts do require considerable functional complexity in the control and monitoring circuitry. The only cost effective way of providing this is via VLSI. This requires that the agency undertaking this work be intimately familiar with all application aspects of this technology.

Through its very complexity, on the other hand, this new level of technology will present new problems of its own. VLS integrated circuits will not be easily testable or controllable unless specific constraints are placed on their design. Subelement testing and control together form the cornerstone of the entire active reliability process. Note that there are tens of thousands of gates and one hundred or so input/output lines on a single integrated circuit. Test vectors long enough for thoroughly testing a random design of this complexity will be far too long to be generated in any reasonable time frame for production testing, much less for fault monitoring in real time. To implement active reliability in systems making extensive use of VLSI requires either the brute force approach or development of a totally new way of designing these components.

At first blush, the brute force approach may sound attractive on the basis of the density and "spare real estate" on a VLSI silicon chip. But system designers are rapidly finding ways of using this new space. In the near future, new systems won't become significantly smaller; they will become more complex. Therefore, the same tension will occur between the additional space, weight, and power required by a fault tolerant concept on the one hand and additional operational functions on the other. The more economical active reliability engineering approaches are mandatory at this level as well as at the others.

The solution to this problem must be addressed at two levels: the chips themselves and the breadboards fabricated from these chips. At the chip level, the chips themselves are the systems and their constituent subfunctions are subelements. The level of complexity represented by these chips justifies (in reality mandates) this definition. At the second level, the breadboard consists of four to twenty chips. This is equivalent to a major system by today's standards. In this case, the breadboard is the system and the chips are the subelements.

At both levels, the solution to this problem lies in application of the same principles as developed for the previous two sections; but the principles must be adapted to each level. The key to this problem is to define basic functional building blocks from which circuits of this complexity are structured. Envisioned by this task is the definition of a library of subfunctions that can be incorporated into automated design processes for these ICs. For each functional building block, an active reliability component subfunction is defined. When these subfunctions are combined in various ways to configure various IC chip designs, an active reliability system subfunction will emerge. Considerable development of these subfunctions is required, as well as considerable work to define them in terms

general enough to be applied to newly defined designs but detailed enough to be useful without extensive further development. These active reliability subfunctions at the chip level will support active reliability at a higher (functional breadboard module) level (fig 7).

To be successful, this program must make one gigantic impact on the IC design process for VLSI chips. The random design process must be eliminated and replaced by an orderly process that builds chips out of subfunctions. This requires that rules for design be established and enforced. The widespread use of Computer Aided Design (CAD) tools in this design practice is opportune while the technology of CAD is in its formative stages. Otherwise CAD will be brought to maturity without active reliability engineering. If it is not among the CAD tools, active reliability will not become integrated into the design process for very large scale (and beyond) ICs. The only alternative of active reliability design at this level—and this level impacts all others—is brute force fault tolerance. But if we do get active reliability into CAD, then every design engineer becomes an active reliability engineer by virtue of his application of the tools.

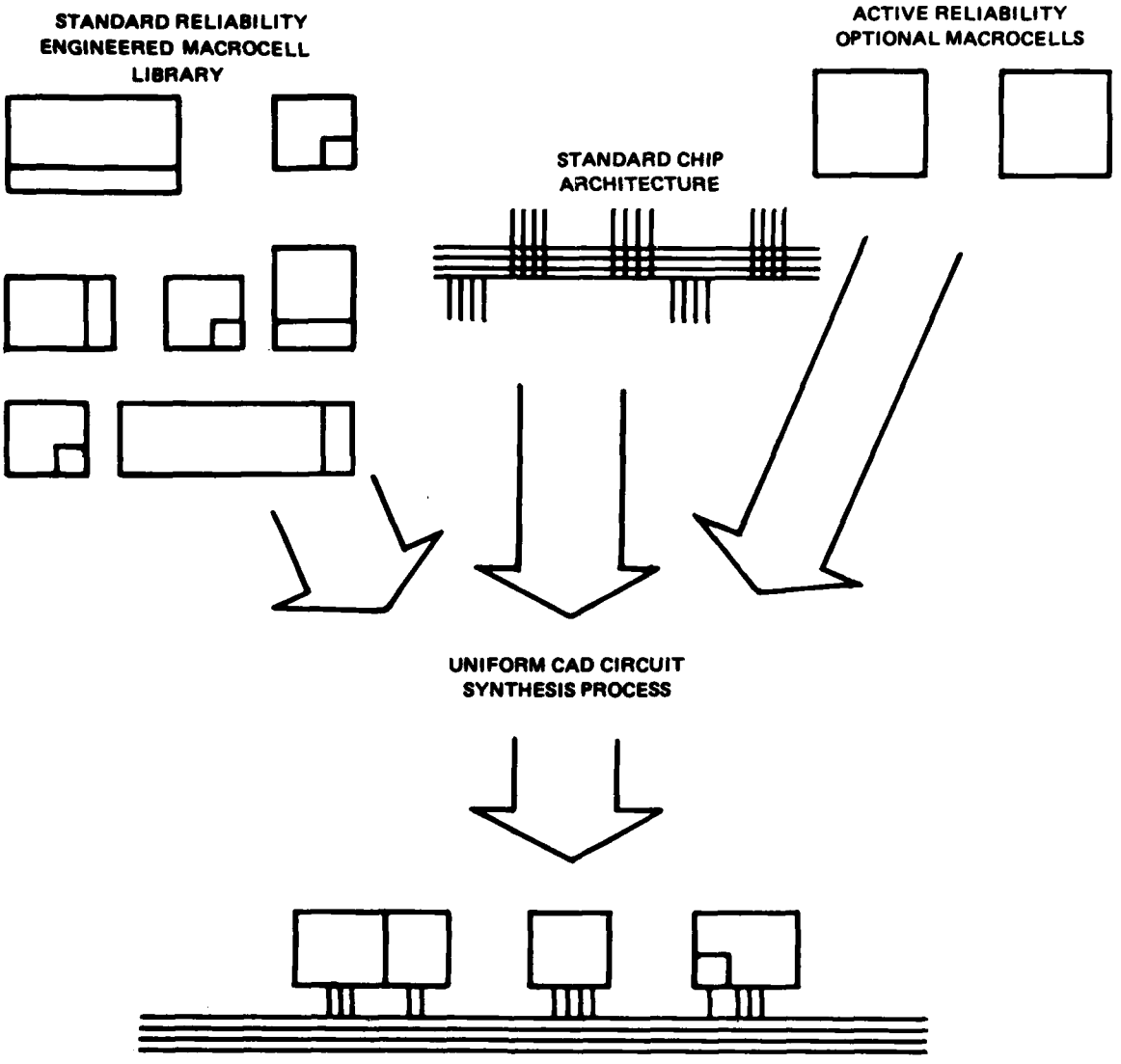


Figure 7. VHSIC active reliability.

6 PROGRAM DESCRIPTION

This section consists of a work breakdown structure. It describes the tasks to be accomplished in prosecuting this program initiative to bring the discipline of active reliability engineering to maturity. Each task is described in detail sufficient to make it possible both to estimate the scope of work required and to develop a general program structure. Full task descriptions and a detailed milestone chart will be developed when the program is initiated.

The level of effort required for each task was estimated on the basis of the level of detail of the task description. These estimates were converted to financial cost estimates by using FY 83 figures for costs of labor and material. The program duration is expected to be 5 years, and the costs estimated are not escalated for the outyears. Based on unescalated costs and a 5-year duration, the total cost of the program would be approximately $24.9M. The costs break out over the 5 years as follows:

FY 84	FY 85	FY 86	FY 87	FY 88
$2500k	$4500k	$6800k	$6200k	$4900k

Major cost factors are those of implementing the test bed and conducting experimental analysis and validation of the concepts developed. These are reflected in the bulge in FY 86 and FY 87 cost figures. These activities are vital to the success of the program.

A breakout of these costs by funding category follows:

6.1	6.2	6.3a	6.3-6.4
Basic Research	Tech Base	Generic	Program Funded
$600k	$4800k	$12 600k	$6900k

The 6.3a (generic) figure represents a funding contribution from a line item that would be directed specifically at this problem. The 6.3-6.4 figure represents a potential contribution from other line item programs that would benefit from this program, such as the Tactical Embedded Computer Program or various combat system acquisition programs. It is proposed that 6.3a funding come from a new line item that could be directed to this problem only or could cover a range of solid-state system problems.

ACTIVE RELIABILITY ENGINEERING PROGRAM

WORK BREAKDOWN STRUCTURE

WBS ELEMENT 100

TASK TITLE: PROGRAM MANAGEMENT

TASK DESCRIPTION:

This task covers the effort and costs associated with the management of this program at NOSC over its 5-year duration.

APPROXIMATE COST: $450k

WBS ELEMENT 210

TASK TITLE: Mathematical Analysis

TASK DESCRIPTION:

Conduct analysis to provide the baseline mathematical-probabilistic models that support the other 200 series tasks. This includes analysis of theoretical or fundamental limits, bounds, and constraints in the application of active reliability techniques. Mathematical models will be derived in the forms of curves describing the effect of redundant hardware on long and short term reliability, statistical models for fault coverage, statistical models for testability, mathematical models for active vs passive trade-offs, and models for software faults. These models are intended to be generic in the sense that they will support a broad range of following tasks in this program and the active reliability engineering process in general, but they also will be sufficiently formalized for inclusion in future military standards and handbooks for active reliability. Some related efforts are ongoing at universities and industry IRAD, and a unified Navy effort to pull together and systematize their results is required.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $160k

WBS ELEMENT 220

TASK TITLE: Evaluation of Existing Reliability Handbooks and Models

TASK DESCRIPTION:

Conduct a detailed analysis of existing (traditional) reliability models as reflected in the existing military reliability handbooks and in standard textbooks. Using the baseline models developed in Task 210, evaluate the effectiveness of the existing models for use in support of active or combined (active and passive) reliability. This analysis will include models of a probabilistic, mathematical, and flow-graph character. Deficiencies will be described, as well as a plan for adapting the existing models.

ANTECEDENT TASKS: 210

APPROXIMATE COSTS: $50k

WBS ELEMENT 230

TASK TITLE: Development of Active/Passive Reliability Models

TASK DESCRIPTION:

In this task, the baseline reliability models necessary for the combined active and passive approaches will be developed. These models shall be compatible with the existing passive models, but shall be modified sufficiently to support cost/benefit trade-offs, active/passive reliability analyses to support contract specification deliverable items, and reliability validation and demonstration. Key attention shall be paid to the role of active redundancy in probabilistic reliability calculations and verification of very reliable systems. This task will achieve a single formal language of reliability modeling by merging the presently diverse definitions associated with active and passive reliabilities.

ANTECEDENT TASKS: 210, 220

APPROXIMATE COST: $120k

WBS ELEMENT 240

TASK TITLE: Development of Generic Cost/Benefit Models

TASK DESCRIPTION:

This task will further develop the baseline reliability models developed in task 230 to the point of generic cost/benefit models. These models will serve as a basis for trade-off analyses by system developers for determining the optimum mix of active and passive approaches to achieve the overall reliability requirement based on the relative cost and payoff of each. This task will take the models developed in previous tasks and reduce them to one or two rule-of-thumb models suitable for use in a designers handbook. The goal is to achieve models that the designer can use to plug in requirements and crank out results. The results of this task will be incorporated in the active reliability handbook.

SUPPORTING TASK: 230

APPROXIMATE COST: $100k

WBS ELEMENT 250

TASK TITLE: Development of Specification and Analysis Procedures

TASK DESCRIPTION:

Use the models developed in Tasks 230 and 240 to develop an initial set of procedures for including a total (active and passive) reliability figure in the system and equipment specifications. Develop the procedures and outline the documentation required to institutionalize a reliability analysis by the contractor.

ANTECEDENT TASKS: 230, 240

APPROXIMATE COST: $215k

WBS ELEMENT 260

TASK TITLE: Development of Verification Procedures

TASK DESCRIPTION:

Conduct an analysis to support the initial development of verification and demonstration procedures for active and total reliability. These procedures may be statistical in nature, and the analysis must address the key problems of visibility into actively reliable systems, assessment of actual fault coverage, components, fault masking, and fault simulation. The procedures developed in this task will be incorporated into a military standard for active reliability by a later task.

ANTECEDENT TASKS: 210, 220, 250

APPROXIMATE COST: $180k

WBS ELEMENT 310

TASK TITLE: Weapon System Requirements Analysis

TASK DESCRIPTION:

The goal of this task is to establish requirements for active reliability in shipboard and avionics weapon systems. This includes both performance and availability figures of merit. Performance requirements will focus on bandwidth, delay time, and data rate constraints resulting from the tactical problem. Availability figures will address issues of mission duration, downtime during mission, etc. This task will concentrate on the unique requirements of weapon systems and their supporting sensors. The requirements for maintenance personnel will be addressed.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $90k

WBS ELEMENT 320

TASK TITLE: Command and Control System Requirements

TASK DESCRIPTION:

The goal of this task is to establish requirements for active reliability for shipboard and avionics command and control systems. This includes both performance and availability figures of merit. Performance requirements will focus on bandwidth, data rate, delay time, and data volume constraints resulting from the tactical problem. Availability figures will address issues of mission duration, downtime during mission, etc. This task will concentrate on the unique requirements of command and control and their supporting systems. The requirements for maintenance personnel will be addressed.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $90k

WBS ELEMENT 330

TASK TITLE: Navy Standard Computer Implications

TASK DESCRIPTION:

Examine the impact of the next generation of Navy standard computers (AN/UYK-43 and AN/UYK-44) on the requirements for shipboard and avionics systems. This includes both the constraints that these computers could place on active reliability concepts (by virtue of their specified reliability and maintenance figures) and requirements that active reliability could place on the computers (either as hardware or software options or upgrades to future versions).

ANTECEDENT TASKS: 310, 320

APPROXIMATE COST: $60k

WBS ELEMENT 340

TASK TITLE: Generalization of Requirements

TASK DESCRIPTION:

Use the results of tasks 310-330 to derive to the maximum extent possible generalized equipment requirements for active reliability. The goal is to define requirements that could be placed on shipboard and avionics equipment in general as well as on a class basis. This task will be general in its statements but is essential for later tasks that define specific design rules or techniques.

ANTECEDENT TASKS: 310, 320, 330

APPROXIMATE COST: $120k

WBS ELEMENT 350

TASK TITLE: Software Issues

TASK DESCRIPTION:

Analyze the requirements placed on software design and software products by active reliability. This task is to cover the requirements to be placed on standard executive software, application software, and software generation tools. The requirements placed on active reliability by software must be examined. For example, limits on software correctness may constrain overall system reliability. This task will set goals and requirements for the follow-on software tasks.

ANTECEDENT TASKS: 310, 320

APPROXIMATE COST: $150k

WBS ELEMENT 360

TASK TITLE: Human-Induced Faults

TASK DESCRIPTION:

This task will examine the problem of human-induced faults in the shipboard and avionics systems from a requirements standpoint. The role of active hardware and software measures to reduce or eliminate the effects of such faults will be examined. This task will establish requirements by examining Fleet user experience.

ANTECEDENT TASKS: 310, 320

APPROXIMATE COST: $75k

WBS ELEMENT 370

TASK TITLE: Analog and Mechanical Components

TASK DESCRIPTION:

Examine the unique requirements deriving from strictly analog or mechanical components in shipboard and avionics systems.

ANTECEDENT TASKS: 310, 320

APPROXIMATE COST: $45k

WBS ELEMENT 410

TASK TITLE: Formal Definition of Active Reliability Parameters

TASK DESCRIPTION:

This task will establish some baseline definitions for active reliability. This list will include active and passive reliability, the approaches to active reliability, the parameters of interest, the role of the human-in-the-loop, etc.

SUPPORTING TASKS: NONE

APPROXIMATE COST: $95k

WBS ELEMENT 420

TASK TITLE: Active Reliability Architecture Classification Definition

TASK DESCRIPTION:

The goal of this task is to develop a systematic scheme for classifying the various architectures—technical approaches and functional configurations—for achieving active reliability. This classification taxonomy will include all existing approaches as well as new ones proposed as part of this task. The classification scheme will be used to find "holes" (potential architectures not yet postulated) in itself and thereby to assure coverage of all possible techniques. Each architecture will be formally defined and described.

SUPPORTING TASKS: 410

APPROXIMATE COST: $75k

WBS ELEMENT 430

TASK TITLE: Active Reliability Architecture Parameterization

TASK DESCRIPTION:

In this task, each architecture defined in Task 420 will be further described in terms of the parameters defined in Task 410. Each architecture will be completely described in terms of these parameters, and an assessment will be made to determine whether this description is complete.

SUPPORTING TASKS: 410, 420

APPROXIMATE COST: $150k

WBS ELEMENT 440

TASK TITLE: Active Reliability Architecture Trade-Off Analyses

TASK DESCRIPTION:

Extensive trade-off analyses will be performed in this task to determine the best architecture(s) for active reliability. The analyses will be based on comparison of performance with respect to the parameters defined in Task 430 and the criterion developed as part of this task. This criterion will address the architectures as they respond to problems of satisfying multiple levels and multitiered requirements, relative cost of implementation, and compatibility with VLSI.

SUPPORTING TASKS: 430, 480

APPROXIMATE COST: $300k

WBS ELEMENT 450

TASK TITLE: Development of Systematic Application Processes

TASK DESCRIPTION:

This task will establish the foundation for the subsequent application tasks. The goal of this task is to develop the general processes of application of the selected architecture. These processes must be broadly applicable and are addressed by this task at a general functional level. This task, however, must address all the methods of application including universal design principles and handbooks, design rules to be incorporated into design tools, definition of functions or functional units to be included in circuit design in general, and unique components or circuits for active reliability.

SUPPORTING TASKS: 240, 250, 440

APPROXIMATE COST: $245k

WBS ELEMENT 460

TASK TITLE: Formalization of Designers Bag of Tricks

TASK DESCRIPTION:

This task will use the preliminary work from Task 450 in the area of design principles and techniques and will formalize them into a designers "bag of tricks" or handbook. The long-term goal is for these guidelines to be validated in the test bed and included in a military handbook. The goal of this task is only to bring the bag of tricks to a state of maturity so as to allow a variety of application designers to experiment with them.

SUPPORTING TASK: 450

APPROXIMATE COST: $120k

WBS ELEMENT 470

TASK TITLE: Active Reliability Design and Analysis Tools (CAD) Definition

TASK DESCRIPTION:

This task will use the results of task 450 to further the preliminary findings of the role of automated design and analysis tools. It will conduct a detailed analysis of the automated design tools to determine what features or design rules for active reliability can be included. The goal is to be able to define procedures or design rules that can be incorporated directly into the tools to make active reliability design integrate with all other phases of the design process through use of the tools.

SUPPORTING TASKS: 250, 450

APPROXIMATE COST: $180k

WBS ELEMENT 480

TASK TITLE: Software Related Concepts

TASK DESCRIPTION:

This task will establish software concepts for active reliability. It will consist of two primary thrusts: the first, software concepts or products to support hardware or system active reliability; the second, the problem of software-induced faults. This task will establish preliminary concepts to deal with the problem of software reliability to be incorporated in the analysis of other tasks in this analysis section. It will also provide preliminary definition of software functions or products that will support and implement active reliability in general.

SUPPORTING TASKS: 350

APPROXIMATE COST: $325k

SUBTASK TITLES: 481—Software Support for Active Reliability
482—Software-Induced Faults Analysis

WBS ELEMENT 490

TASK TITLE: Human-Induced Fault Analysis

TASK DESCRIPTION:

This task will provide analyses and recommendations to solve the problem of human-induced faults. The approach must satisfy the requirements derived in Task 360.

SUPPORTING TASK: 360

APPROXIMATE COST: $80k

WBS ELEMENT 510

TASK TITLE: Testability Impacts/Inputs Analysis

TASK DESCRIPTION:

The goal of this task is to produce outputs that will drive research in the testing community in the area of testability and built-in test (BIT). This task will accomplish three things: recommendations and direction to the test technology research problem, direct investigation of testability-related issues, and recommendations for additional testability work. This task provides the interface to the testing community and is needed because self-checking and monitoring are vital functions supporting active reliability, there will be significant commonality between the two disciplines, and fault tolerance can, in some cases, interfere with testability by masking faults. In addition, the problem of statistical estimating of testability effectiveness and fault coverage is required to support active reliability analysis and design. This area will require theoretical as well as practical empirical analyses. A level of confidence must be ascribed to a given level of self-testing, so that overall reliability estimates can be developed.

ANTECEDENT TASKS: 210, 230

APPROXIMATE COST: $350k

WBS ELEMENT 520

TASK TITLE: On-Line Maintenance Procedure Analysis

TASK DESCRIPTION:

The object of this task is to develop and formalize a series of possible approaches to on-line maintenance that will be compatible with active reliability. This task will examine the problem of maintenance and repair of system components while the system maintains an active or standby status. The problems of isolation of components, automated diagnostic and repair aids, notification of operators, and safety will all be addressed. The key issue is the establishment of initial procedures to integrate active reliability/automated repair with a system maintenance philosophy and level-of-repair evaluation.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $215k

WBS ELEMENT 530

TASK TITLE: Redundancy Management (Configuration/ILS)

TASK DESCRIPTION:

Develop strategies for automated management of redundant resources. Two distinct strategies are required. One consists of the establishment of rules for automated configuration from a stock of spare resources. Management implies the ability to keep track of the status and availability of hot and cold spare resources in the on-line system environment. The other strategy must address the management of spare resources beyond the confines of the system (i.e., in the part cabinet and ILS chain). The availability of a system is limited in the long run by the availability of spares. Automated ILS management is required, to track spares availability.

SUPPORTING TASKS: NONE

APPROXIMATE COST: $250k

WBS ELEMENT 540

TASK TITLE: Redundancy Control Techniques

TASK DESCRIPTION:

Develop further the strategy for control of redundant resources (on-line hot and cold spares). The objective of this task is to advance this work to the point of detailed control procedures and control protocols. (The previous task develops a general strategy for system configuration control.) This task is aimed at deriving generic but detailed procedures for exerting control over redundant resources.

ANTECEDENT TASK: 530

APPROXIMATE COST: $250k

WBS ELEMENT 550

TASK TITLE: ATE/On-Line Performance Monitoring Analysis

TASK DESCRIPTION:

Evaluate the impacts and relationship between active reliability and the off-line automatic test equipment (ATE) and on-line performance monitoring environments. This analysis will address the problems of detailed testing of the active and inactive paths of the redundant system, on-line performance monitoring scans of the redundant resources, and standard interfaces with off-line ATE. This task will drive ongoing work on the ATE systems and on-line performance monitoring technologies by the testing technology community.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $175k

WBS ELEMENT 610

TASK TITLE: UYK-44 Network Active Reliability Concept

TASK DESCRIPTION:

Develop an active reliability concept for large scale systems based on networking of UYK-44 computers. This approach is to be directed toward development of this concept for the network as a whole rather than just for the UYK-44 units themselves. It is to be based on minimal modifications or purchase options consisting of hardware or software functional units. The concept must address the computers, network data transmission and interface units, and peripheral devices. The work done under this tasking will consist of analyzing the UYK-44 and its network applications relative to the requirements established in previous tasks and applying the active reliability functional architecture derived in Task 440. Functional requirements will be established to direct the subsequent 600-series tasks. This approach must be flexible enough to address the full range of networks covered by the UYK-44. A cost-benefit analysis will be conducted.

ANTECEDENT TASKS: 230, 240, 330, 440

APPROXIMATE COST: $300k

WBS ELEMENT 620

TASK TITLE: UYK-44 Network Architecture/Protocol Impacts

TASK DESCRIPTION:

Evaluate the impacts of a network active reliability concept on the network architecture and data transfer protocols. This process will consist of analysis of the active reliability concept developed in Task 610 and the network architecture and interface work being developed by UYK-44 based acquisition programs. Requirements/ modifications to the network architectures and additional network protocols will be promulgated to the UYK-44 community. These protocols will be developed in detail down to the bit level. They will address all aspects of intranetwork communication including command messages and configuration control, intranetwork testing, and status/error reporting.

ANTECEDENT TASKS: 610

APPROXIMATE COST: $265k

WBS ELEMENT 630

TASK TITLE: UYK-44 Hardware Modifications

TASK DESCRIPTION:

Develop detailed functional descriptions of hardware modifications to the UYK-44 required to implement network active reliability. These modifications are not intended to be made to the baseline UYK-44 computer or its circuit cards. They will consist of hardware purchase options in the form of additional SEM format module cards. The goal of this task is not the detailed design and fabrication of these circuits, but rather their preliminary functional design in a form sufficient for incorporation in detailed procurement specifications. Preliminary VLSI designs will be made where required. Cost estimates will be provided.

ANTECEDENT TASKS: 610, 660

APPROXIMATE COST: $350k

WBS ELEMENT 640

TASK TITLE: UYK-44 Software Modifications

TASK DESCRIPTION:

Develop detailed functional descriptions for software modifications to achieve network active reliability. These modifications will concentrate on alterations to the Navy standard minicomputer executive program, SDEX-M, and definition of application program modules. These functional units are to be described at the program performance specification level. The software modifications must be compatible with the hardware options developed in Task 630, but in some cases they will be duplicative in order to give the designer a hardware/software option.

ANTECEDENT TASKS: 610, 620, 630

APPROXIMATE COST: $375k

WBS ELEMENT 650

TASK TITLE: Ultrareliable UYK-44 Development

TASK DESCRIPTION:

Develop one or more highly reliable and ultrareliable versions of the UYK-44 either as modifications to existing UYK-44 computers or upgrades to future versions. In this task, tentative architectures will be advanced, including multiple UYK-44 units with voting, addition of self-checking features and total reengineering. These architectures will be proposed in detail and evaluated for their relative effectiveness. Evaluation factors will include cost, applicability to network and other requirements, and adaptability to a variety of requirements. A detailed functional design of the optimal architecture will be provided.

APPROXIMATE COST: $335k

WBS ELEMENT 660

TASK TITLE: Resource/Peripheral Switching Issue Resolution

TASK DESCRIPTION:

Develop an approach for the control and switching of peripheral devices and other network-related resources and the UYK-44 computers comprising the network. The network active reliability architecture provides for switching among redundant UYK-44s and the associated transfer of functions. In addition, the ability to switch external resources (tapes, disks, displays, etc) and controlled units (sensors, weapons) among the computers is required. This task will define a scheme compatible with the overall network scheme advanced in Task 610 and will meet all of the objectives set forth therein. A function description of the switching philosophy and devices will be provided.

ANTECEDENT TASKS: 610

APPROXIMATE COST: $275k

WBS ELEMENT 670

TASK TITLE: UYK-43 Active Reliability Upgrade

TASK DESCRIPTION:

Develop recommendations for functional capabilities to be added to the UYK-43 as part of a midlife technology upgrade to that computer. It is anticipated that a midlife upgrade to the UYK-44 will be conducted in place by replacement of existing circuit boards with new boards designed with higher levels of circuit integration. These boards will allow for addition of more functional capability to the upgraded UYK-43. This task will propose additional functions to be included to enhance the existing capabilities for fault tolerance contained in the UYK-43 to a fully active reliability concept.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $285k

WBS ELEMENT 680

TASK TITLE: UYK-44 Network Active Reliability Demo

TASK DESCRIPTION:

Implement the active reliability network concept in a test bed environment, using the test bed developed in the 800-series tasks. Apply to a simulated problem. Conduct exercise for demonstration and evaluation purposes. This task includes the development and fabrication of prototype circuit cards and construction of a model network.

ANTECEDENT TASKS: 630, 640, 660

APPROXIMATE COST: $3.2M

WBS ELEMENT 710

TASK TITLE: Functional Design Concept Analysis

TASK DESCRIPTION:

Conduct the analysis necessary to develop an active reliability design concept based on the use of multipurpose functional modules. In this task, the concept will be defined and an investigation will be undertaken to determine/establish areas of its applicability. Basic functional units will be defined. An active reliability concept will be defined and applied to the units. A preliminary scheme for building large functions from the units and for interfacing them will be developed. The applicability of VLSI/VHSIC technology to this approach will be shown.

ANTECEDENT TASKS: 230, 240, 440

APPROXIMATE COST: $350k

WBS ELEMENT 720

TASK TITLE: Functional Design Concept Interface Definition

TASK DESCRIPTION:

Rigorously define the interface procedure and signals to achieve interface between the functional units. This includes methods of common data transfer, control and status signals, and redundant control strategy.

ANTECEDENT TASK: 710

APPROXIMATE COST: $320k

WBS ELEMENT 730

TASK TITLE: Functional Design Concept Unit Definition

TASK DESCRIPTION:

Further the definition of the basic functional unit(s) to the point of detailed design suitable for implementation.

ANTECEDENT TASKS: 710, 720

APPROXIMATE COST: $575k

SUBTASK TITLES: 731—Functional Microcomputer Unit

732—Functional Signal Processor Unit

733—Functional Interface & Control Unit

WBS ELEMENT 740

TASK TITLE: Functional Design Concept Packaging Definition

TASK DESCRIPTION:

Determine a standard packaging technique compatible with existing hardware standards and addressing the requirements of the functional design concept.

ANTECEDENT TASK: 730

APPROXIMATE COST: $325k

WBS ELEMENT 750

TASK TITLE: Functional Design Concept VLSI Application/Demonstration

TASK DESCRIPTION:

Implement the functional design concept to use VLSI circuitry and apply it to an exemplary problem. Exercise the application on the test bed for demonstration and evaluation purposes. Use the demo and test bed to aid in the transfer of this design process to industry.

ANTECEDENT TASKS: 710, 730, 740

APPROXIMATE COST: $2.75M

WBS ELEMENT 760

TASK TITLE: VLSI/VHSIC Active Reliability Concept

TASK DESCRIPTION:

Select one area of interest for application of the design concept to microcircuitry. Extend the functional design concept down to the design of individual VLSI and VHSIC modules.

ANTECEDENT TASK: 730

APPROXIMATE COST: $450k

WBS ELEMENT 770

TASK TITLE: VLSI/VHSIC Design Rules Definition

TASK DESCRIPTION:

Formalize the VLSI/VHSIC concept for transition to industry and universal application, by means of design rules.

ANTECEDENT TASKS: 460, 760

APPROXIMATE COST: $255k

WBS ELEMENT 780

TASK TITLE: VLSI/VHSIC CAD Modification

TASK DESCRIPTION:

Demonstrate the application of design rules for VLSI/VHSIC active reliability by modifying a CAD system to incorporate the rules.

ANTECEDENT TASKS: 470, 760, 770

APPROXIMATE COST: $1.85M

WBS ELEMENT 790

TASK TITLE: VLSI/VHSIC Implementation/Demo

TASK DESCRIPTION:

Implement the CAD design rules for VLSI/VHSIC active reliability developed in task 780 and apply them to an exemplary problem. Implement the resulting components in the test bed for exercise and demonstration purposes.

ANTECEDENT TASK: 780

APPROXIMATE COST: $2.15M

WBS ELEMENT 795

TASK TITLE: Power Source Active Reliability

TASK DESCRIPTION:

Investigate the issues of primary and secondary power source reliability in an active reliability environment. Propose an approach to make power source reliability compatible with system reliability figures to be achieved by active reliability engineering. Validate this concept and translate it into design requirements or specification language as required.

ANTECEDENT TASK: 340

APPROXIMATE COST: $1.7M

WBS ELEMENT 810

TASK TITLE: Test Bed Requirements and Functional Description

TASK DESCRIPTION:

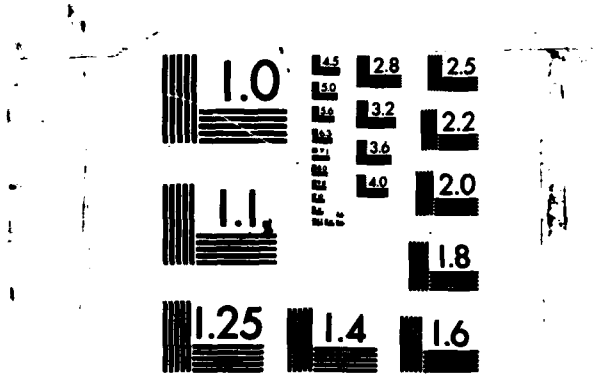
Develop a formalized definition of the test bed requirement and the test bed implementation. Write this up in a functional description document.

ANTECEDENT TASKS: NONE

APPROXIMATE COST: $225k

AD-A137 541 ACTIVE RELIABILITY ENGINEERING - TECHNICAL CONCEPT AND PROGRAM PLAN A SOL..(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA D D HALL 05 OCT 83 NOSC/TD-654 2/2

UNCLASSIFIED F/G 9/2 NL



WBS ELEMENT 820

TASK TITLE: Test Bed Design

TASK DESCRIPTION:

Provide a formalized design of the test bed to meet the requirements of the functional description and goals of the program.

ANTECEDENT TASKS: 610, 710, 810

APPROXIMATE COST: $350k

WBS ELEMENT 830

TASK TITLE: Test Bed Acquisition

TASK DESCRIPTION:

This task covers the acquisition of components for the test bed.

ANTECEDENT TASK: 820

APPROXIMATE COST: $650k

WBS ELEMENT 840

TASK TITLE: Test Bed Development

TASK DESCRIPTION:

This task provides the development of the test bed, including test and evaluation software, according to the previously developed design.

ANTECEDENT TASKS: 820, 830

APPROXIMATE COST: $1.2M

WBS ELEMENT 850

TASK TITLE: Test Bed Operation and Maintenance

TASK DESCRIPTION:

This task covers the operation and maintenance costs of the test bed for the duration of the program, but not the cost of placing experiments on it.

ANTECEDENT TASK: 840

APPROXIMATE COST: $900k

WBS ELEMENT 910

TASK TITLE: Active Reliability Design Guide

TASK DESCRIPTION:

Develop and update a design guide as a formal document to assist design and acquisition engineers in the application of active reliability concepts to their systems.

ANTECEDENT TASKS: 230, 240, 460, 795

APPROXIMATE COST: $250k

WBS ELEMENT 920

TASK TITLE: Active Reliability Military Standard

TASK DESCRIPTION:

Develop and update a military standard for active reliability.

ANTECEDENT TASKS: 250, 260, 460

APPROXIMATE COST: $300k

WBS ELEMENT 930

TASK TITLE: Active Reliability Military Handbook Development

TASK DESCRIPTION:

Develop and update a military handbook to incorporate data from the design guide as the mechanism for describing formal procedures and formal technical language for the interchange of analysis data between the contractor and the acquisition manager.

ANTECEDENT TASKS: 250, 910

APPROXIMATE COST: $300k

WBS ELEMENT 940

TASK TITLE: Active Reliability Specification Language

TASK DESCRIPTION:

Develop and update the language for detailed specification of active reliability including data item descriptions, analysis documentation procedures, and verification procedures.

ANTECEDENT TASKS: 250, 260, 795

APPROXIMATE COST: $150k

END

FILMED

3-84

DTIC